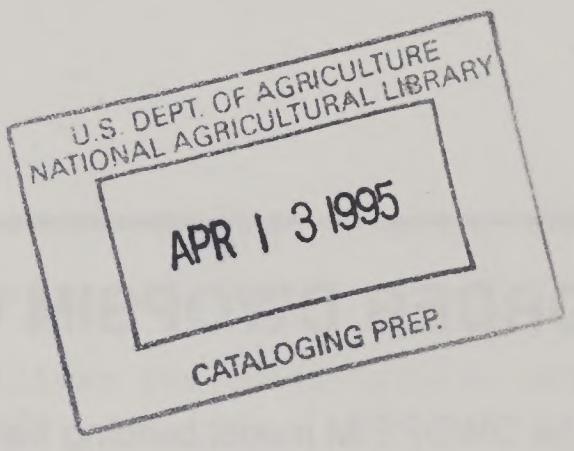
tates
ient of
ure

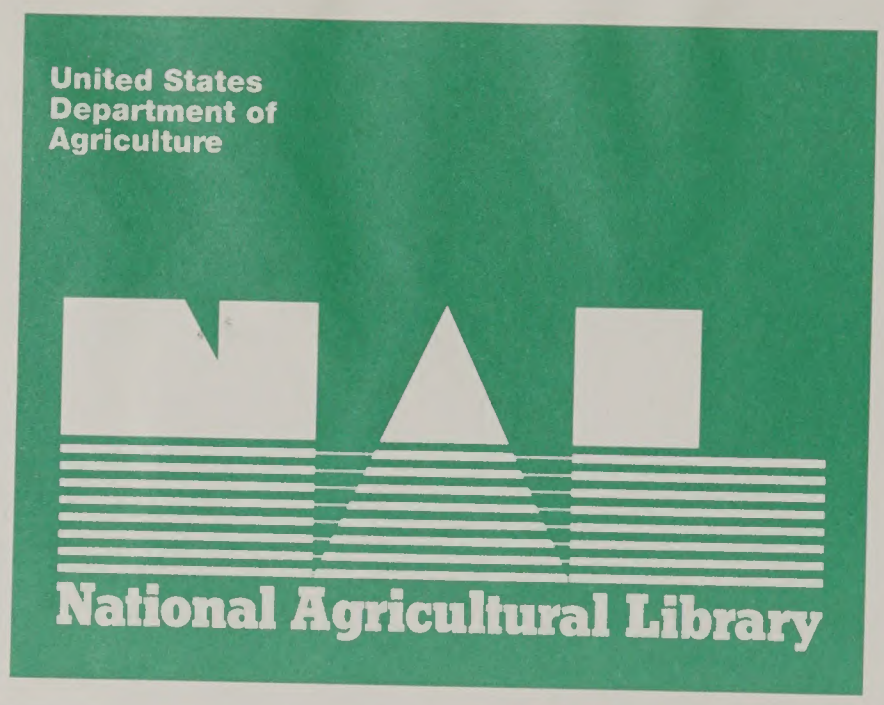
nic
ch

Agriculture
and Trade
Analysis
Division

Documentation of the Dynamic World Policy Simulation (DWOPSIM) Model Building Framework

Vernon Oley Roningen





ORDER DWOPSIM ON DISK

The DWOPSIM model building framework and the sample model TIME are available on a 3.5", 1.44 MB diskette. Please note that DWOPSIM requires an IBM or compatible computer, DOS 3.3 or higher, SuperCalc 5 spreadsheet software, 2 MB of available RAM, and at least 3 MB of hard disk space. Also, the TSView database used by DWOPSIM (see Appendix 1) is available on two 3.5" diskettes. This product requires 330 KB of RAM and at least 650 KB of hard disk storage.

To order, just dial 1-800-999-6779. Toll free in the United States and Canada. Other areas, please call 1-703-834-0125.

Ask for the *DWOPSIM* database, order #93001, $25, and the *PS&D View* database, order #93002A, $35. Add 25 percent for shipping to foreign addresses (including Canada). Charge your purchase to your VISA or MasterCard. Or send a check (made payable to ERS-NASS) to:

ERS-NASS
341 Victory Drive
Herndon, VA 22070

For further information on this database, write to Vernon Roningen, Room 735, 1301 New York Avenue, NW., Washington, DC 20005-4788 or call (202) 219-0682.

Documentation of the Dynamic World Policy Simulation (DWOPSIM) Model Building Framework. By Vernon Oley Roningen, Agriculture and Trade Analysis Division, Economic Research Service, U.S. Department of Agriculture, Staff Report No. AGES 9226.

Abstract

This report documents the dynamic world policy simulation (DWOPSIM) model building framework. DWOPSIM, a set of DOS batch computer programs and spreadsheet templates, allows users to assemble globally consistent data sets into time series models. Such models are used for policy analysis, short-term forecasting, and long-term projections. Users choose equations and code the structure along with model parameters into spreadsheet templates. DWOPSIM provides computer programs for the generation of model equations, the calculation of constant terms, the assembly of spreadsheet components into a linked global model, and the generation of simulation output.

Keywords: Agricultural trade, DWOPSIM, forecasting model, projections model, trade, trade model, world trade model.

Acknowledgments

I would like to thank Steve Haley and John Sullivan for their helpful review comments and Bonnie Moore for her editorial help.

1301 New York Ave., NW
Washington, DC 20005-4788

October 1992

Contents

Documentation of the Dynamic World Policy Simulation (DWOPSIM) Model Building Framework

Vernon Oley Roningen

Introduction

This report documents the dynamic world policy simulation (DWOPSIM) model building framework. DWOPSIM allows the user to construct global multi-product, multicountry time series models and solve them in spreadsheets. The framework is a set of DOS batch computer programs and spreadsheet templates that help a user with various tasks needed to build time series models, including the generation of model equations from a model specification coding scheme, the calculation of constants/intercepts for equations for a base period, the assembly of product and country spreadsheets into a linked world model, and the generation of simulation output. The framework also includes routines that construct globally consistent supply and utilization tables from a TSView database. The use of the DWOPSIM framework is illustrated with a small two-product, three-region world model that forecasts/projects variable values from the present to the year 2020.

The report begins with an overview of the DWOPSIM model building framework and a flowchart summarizing the model building process. This is followed by a discussion of data preparation, including a description of DWOPSIM programs. Next, the process of specifying model equations and entering their parameters into DWOPSIM equation spreadsheets is explained. The entry of exogenous data into product, country, and world spreadsheets is then discussed. Procedures for product model initialization and world model assembly are illustrated. Finally, model simulation and output generation are discussed.

The DWOPSIM framework is illustrated with a small global time series model named TIME. Appendices describe the TSView database used for the demonstration model and provide sample documentation of the demonstration model TIME. DWOPSIM computer programs are also documented in an appendix.

Overview of the DWOPSIM Model Building Framework

While static global models are useful and practical for global policy analysis (1,6) and projections (5), there are also good reasons to use models that show the time path of variables under various simulation scenarios.[1] For example, an analyst might want to do short-term forecasting or long-term projections. Dynamic considerations represented by lagged variables might be considered important in certain modeling situations. From a technical viewpoint, it is more difficult to create well-behaved dynamic time series models than static models. Data requirements for time series models are greater, and dynamic properties may cause solution problems for such models (4). Just as spreadsheets have proved to be a practical vehicle for the construction of global static models, the DWOPSIM spreadsheet modeling framework can provide similar advantages for dynamic time series-based models.

---

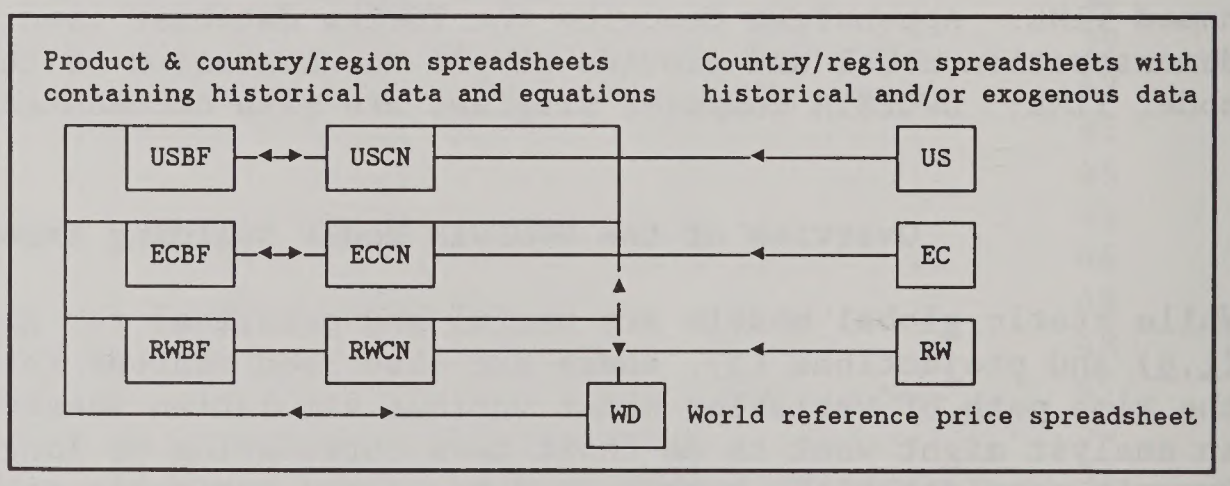
[1]Underscored numbers in parentheses refer to items cited in the References section.

Experience with the static world policy simulation (SWOPSIM) modeling framework (6) led to the DWOPSIM framework, which allows for dynamic time series modeling but retains the spreadsheet as the modeling vehicle. In the static SWOPSIM modeling framework, a premium was put on a standard simple economic structure that allowed much of the routine model building and assembly work to be done by computer programs operating on spreadsheets. Dynamic modeling requires many more choices to be made regarding model specification. Therefore, DWOPSIM was designed to allow the user full discretion in model specification and parameterization, while retaining the convenience and user accessibility of spreadsheets.[2] In addition, given the existence of consistent global data sets available with USDA's TSView database system (2), the DWOPSIM framework includes an option that allows the user to construct a globally consistent supply and utilization data set for products to be modeled. DWOPSIM uses the SuperCalc 5 spreadsheet.[3]

DWOPSIM model components reside in spreadsheets. Organization of the model structure in spreadsheets is entirely up to the user. Because the demonstration model TIME is based on a TSView product data set that is organized by product, each product in TIME resides in a separate spreadsheet for each country. Each country in TIME was given a separate spreadsheet for country specific data. Finally, a world model requires a spreadsheet containing world reference prices and any other global data needed. Figure 1 shows the DWOPSIM data structure selected for the demonstration model TIME used in this report. Following the country and product nomenclature from the TSView data source (7), the model TIME divides the world into three countries/regions, the United States (US), the European Community-12 (EC), and the rest of the world (RW). Two products are modeled, beef and veal (BF) and corn (CN). The model TIME is organized into six product/country spreadsheets, three country/region spreadsheets, and one world reference price spreadsheet. Typical linkages between spreadsheets via model equations are shown with lines and arrows.

The construction of a DWOPSIM model consists of the assembly of data and the coding of the equation structure and parameters in the

Figure 1--Organization structure of spreadsheets for the demonstration model TIME



---

[2]This is different from the SWOPSIM modeling strategy where a simple standard specification was used and built into the SWOPSIM framework. In contrast, DWOPSIM models require the user to provide all of the structure, specification, and parameters for model equations. The user can choose the spreadsheet structure for the model. The demonstration model TIME is organized with one spreadsheet per product; however, alternative organizational schemes are possible.

[3]SuperCalc is a registered trademark of Computer Associates International Inc., 1240 McKay Drive, San Jose, CA 95131, 408-432-1727.

appropriate spreadsheets, the writing of model equations in spreadsheets from the specification codes, the initialization of equations to base period data by the calculation of constant terms, and the creation of a global model by linking the spreadsheets in memory.[4] Much of this construction is carried out by DWOPSIM programs. The updating of data or parameters simply requires a repetition of DWOPSIM operations after manual changes have been made in spreadsheets. The model-building flowchart on the following page summarizes the operations required to build and maintain/update a DWOPSIM model.

The first task in creating a DWOPSIM model is to assemble globally consistent data sets for the products and countries to be modeled in a chosen spreadsheet layout. This task can be done manually. But, if agricultural products from U.S. Department of Agriculture's (USDA), Foreign Agricultural Service (FAS) data are being modeled, the framework provides a set of programs that allow the user to easily prepare a set of quantity supply and utilization data from a SWOPSIM version of USDA's TSView product data (7). Given that the user selects a product from the TSView data set, programs automatically generate a product spreadsheet for each country in a model.[5] This step is accomplished by the program DWOPTIME shown in a box at the top of the flowchart. As with SWOPSIM models, each DWOPSIM model requires a rest-of-the-world region to close world markets for the selected product. DWOPSIM programs adjust data for the closure region (RW) to maintain world identities of zero net trade and balanced world supply and demand in each historical period.

The DWOPSIM model builder must specify and parameterize model equations. This is done by coding in the equation structure and parameters into a specific equation spreadsheet. The user is free to specify any equation structure and must rely on external analysis for equation parameters. In the demonstration model, TIME shown in the flowchart, spreadsheets containing equation codes and equations are organized by product with one country per product sheet.

Once spreadsheets have been created for all products and countries and the associated equation spreadsheets have been coded, the user must add exogenous data required by the model specification. In the example TIME, product-specific data goes into the product (such as USBF) spreadsheet, while country-specific data goes into the country spreadsheet (such as US). In addition, world reference price data must be put into world reference price spreadsheet (WD). Next, the program DWOPINIT is used to initialize the model spreadsheets. This means that information in the equation specification spreadsheet on equation form and parameters is used to write equations and initialize each equation to base period data. When the process has been completed for each spreadsheet, the user is ready to assemble a world model.

---

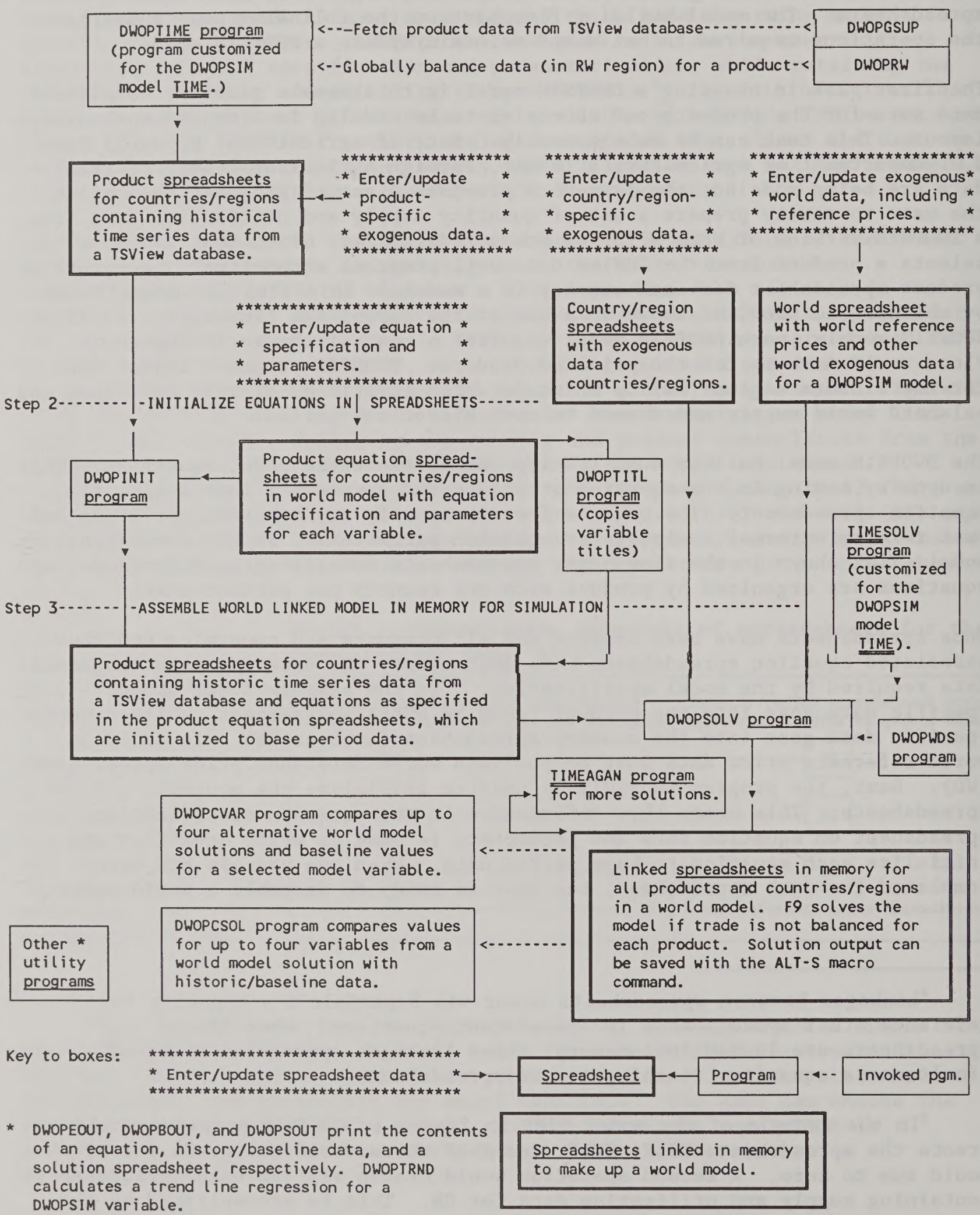
[4]Linkages between spreadsheets occur via SuperCalc 5's capacity to reference other spreadsheets in spreadsheet equations. When linked spreadsheets are loaded into memory, these linkages operate as a set of simultaneous equations reaching across spreadsheets.

[5]In the example of the model TIME in figure 1, DWOPSIM programs would create the spreadsheets USBF, ECBF, and RWBF where world net trade for BF would sum to zero. A second operation would create similar data spreadsheets containing supply and utilization data for CN. This is accomplished by the program DWOPTIME, which has been created and customized for this operation for the model TIME.

DWOPSIM Model Building Flowchart

(the model TIME with a product-based spreadsheet structure used as an example)

Step 1--ASSEMBLE GLOBAL DATA SET (illustrated by the model TIME, which uses a product spreadsheet organization)



Key to boxes:

* DWOPEOUT, DWOPBOUT, and DWOPSOUT print the contents of an equation, history/baseline data, and a solution spreadsheet, respectively. DWOPTRND calculates a trend line regression for a DWOPSIM variable.

The assembly of a world model for simulation is done by the DWOPSIM program DWOPSOLV. The equation portion of product and country spreadsheets is loaded into computer memory along with a world price equilibrium mechanism. When a year's (or years') data are not in balance due to policy or other economic shocks entered into the model, the set of spreadsheets with cross-product and cross-country linkages can be solved in memory to achieve zero world net trade in each simulation period. The solution values of the spreadsheets can be saved and compared with historical/baseline data.

In addition to the DWOPSIM routines mentioned above, various utility programs are available to the user for various tasks, such as printing data and equations. DWOPSIM programs are DOS batch programs that write and execute macros that operate on spreadsheet templates. The casual DWOPSIM user must have some knowledge of DOS and familiarity with SuperCalc 5 spreadsheet commands. However, the mechanics behind DWOPSIM programs are fairly simple. The ambitious user who knows DOS batch programming and SuperCalc 5 macro-commands can easily create DWOPSIM-like routines for his own purposes.

Because dynamic models are more complex and can be specified in many ways, DWOPSIM leaves specification questions largely up to the model builder. The demonstration model (TIME) used to illustrated the DWOPSIM framework builds upon the simple specification and parameters for BF and CN found in the US, EC, and RW model components of the SWOPSIM model DEMO described in SWOPSIM documentation (6). The product-based organization of TIME follows from the product organization of TSView supply and utilization data, but other model organization schemes are possible. Dynamic models typically require more data and parameters than static ones, especially if lag structures are involved. Therefore, questions of model size and computational efficiency become relatively more important for dynamic models than for static models.

Computer Requirements

The technical computer limit for the size of DWOPSIM models is the expanded memory available for holding SuperCalc 5 spreadsheet(s). In addition, SuperCalc 5 has a limit of 255 linked spreadsheets in memory at one time.[6] This means that a microcomputer with a large memory should be able to handle a large model. However, the user is always cautioned by the rule, "big models can be big trouble." Large models are typically slower to solve and more difficult to interpret; thus, there may be "practical" limits to model size that might be reached even if technical computer limitations are not breached. SuperCalc 5 spreadsheet defaults must be set correctly. This can be done automatically by typing in the program name DWOPSIM and by following the directions on the screen.

---

[6]The model TIME used to illustrate DWOPSIM in this report consists of two products (BF and CN) and three regions (US, EC, and RW). This gives 2 x 3 = 6 product/country spreadsheets and one world reference price/solution mechanism spreadsheet, for a total of seven spreadsheets that will have to be kept simultaneously in memory in a linked mode. Three country/region spreadsheets contain only exogenous data so they do not have to be kept in memory. If another organizational scheme was used, such as a country one (where all product data for a country is on one spreadsheet), only four spreadsheets would have to be kept in memory.

The current version of DWOPSIM does have practical limits built into it. For example, the equation-writing program (a spreadsheet itself) allows only 28 variables per model spreadsheet, variable lags of up to 9 periods, and a data (annual) range from 1960 to 2020. These "design" limits could be exceeded with bothersome changes in existing DWOPSIM programs and accompanying spreadsheet templates. These limits are partially based on practical matters, such as a desire to limit model printouts to two pages. Default printouts of DWOPSIM output are dependent upon the capability of laser printers to produce small print so that a lot of information can fit on a page. However, DWOPSIM programs also have the option to print to a disk file so that other programs can "fit" the data to paper.[7]

The programs which take data from the TSView system assume that TSView files are on the E: disk. However, the controlling batch software can be changed for another disk configuration. TSView data are organized as described in appendix 1. TSView data sets with different organization and product structures could also be used, provided that some programming was done to develop suitable two-letter country and product codes (which are required for DWOPSIM equation-writing programs).

Most DWOPSIM programs are applicable to all models, but a few key ones must be constructed for EACH specific model (for example, the programs with TIME included in their name in the model-building flowchart). When the user creates new models, one way of supplying the "model" specific programs and spreadsheet templates is to copy them from the model TIME to a new model name and edit them manually. The DWOPSIM program TIMEREPL (screen 1) gets the user started in this direction by copying the necessary files to the D: drive and renaming them with the new model name. Model configuration and nomenclature in this report (that is, country and product codes) follow those used in the TSView database for SWOPSIM models. DWOPSIM batch programs (with the first four letters of their name being DWOP) are housed in the C:\BATCH subdirectory.

Screen 1--The program TIMEREPL

TIME Replication Program for a New DWOPSIM Model	
TIMEREPL	Program to REPLicate model specific TIME model programs and spreadsheet templates files for use with a new DWOPSIM model, NAME. A two product, three region demonstration model, TIME, comes with the DWOPSIM model building system. NAME*.BAT and NAME*.CAL files will be left on D: and must be edited manually to accommodate the product and country coverage of the new model NAME. *.CAL and *.SUB files must be copied to the C:\NAME subdirectory for safekeeping. *.BAT files must be copied to the C:\BATCH subdirectory.
REQUIREMENTS	The TIME model must exist on the C:\TIME subdirectory and all of the appropriate batch files must exist on the C:\BATCH subdirectory.
OUTPUT (D:)	The output files will reside on the D: subdirectory. These include DWOPNAME.BAT, NAMERW.SUB, NAMESOLV.BAT, and WDS.CAL (which must be customized for products in NAME).
COMMAND	TIMEREPL NAME

[7]Small print is obtained on laser printers in DWOPSIM output programs by invoking Laser Control software. Laser Control is a registered trademark of Insight Development Corporation. Printouts of ASCII files on disk can be obtained in various sizes and configurations of print by the SIDEWAYS program, which comes with the SuperCalc 5 spreadsheet package.

Batch programs specific to the demonstration model TIME (first four letters of their name being TIME) are also housed in the C:\BATCH subdirectory. DWOPSIM programs are kept in the C:\DWOPSIM subdirectory, while spreadsheets and other files for the model TIME are kept in the C:\TIME subdirectory. In contrast to SWOPSIM, DWOPSIM file names do not contain the model name. As was the case with SWOPSIM, DWOPSIM programs leave their results on the D: drive where they must be manually copied to the model subdirectory for permanent safekeeping.

DWOPSIM programs all have explanatory screens (such as screen 1), which can be invoked by typing the program name (use capital letters) without the parameters required by the program. Error traps have been built into the programs to alert the user if file preparation steps have been omitted.

Assembly of Globally Consistent Quantity Data in Model Spreadsheets

The first step in creating a global trade model is the assembly of a consistent set of quantity data, including globally balanced trade data. This can be done manually by carefully entering data into spreadsheets. The number and organization of spreadsheet components of a model is up to the user. It should be kept in mind that a world market-clearing mechanism will have to be designed based on some principle. The model TIME does this by clearing world net trade for each product with a mechanism that changes world prices until net world trade equals zero.

The SWOPSIM modeling framework has routines that reformulate USDA's global FAS supply and utilization data into a TSView 143-country/region database.[8] This database serves as the repository for product supply and utilization quantity data used to generate the product spreadsheet data for each country in the model TIME. The program DWOPTIME (screen 2) is created for the model TIME to automatically carry out the data retrieval operation. DWOPTIME not only gets data for the US and the EC and calculates data for RW (= World - US - EC), but it also performs an exercise on RW data to make sure the total world trade for all

Screen 2--The program DWOPTIME

```
DWOPSIM Program to create TSView-based data for model TIME
-----------------------------------------------------------------------
DWOPTIME       Program to make a complete set of balanced supply and
               demand spreadsheets for a selected PRoduct from a TSView
               data set for each CountrY in the model TIME, including a
               globally balanced RW (Rest-of-World bal. with DWOPBLRW).
REQUIREMENTS   A E:\TS subdirectory must contain the PRoduct TSVIEW files.
               A subtraction control file for TSView software must be on
               the model NAME subdirectory with the name NAMERW.SUB.
               An example of such a file for a three region world (US, EC,
               and RW) where RW = WD - US - EC might be:
               ---------------------------------------------------------
               TEMP.TS
               TEMP.TS
               1
               RW                        1  3  5
                                 RW = WD-US-EC (Codes for TSView #s)
               ---------------------------------------------------------
OUTPUT (D:)    CYPR spreadsheets for each CountrY in TIME plus the
               globally balanced RWPR spreadsheet.
-----------------------------------------------------------------------
COMMAND        DWOPTIME PR
-----------------------------------------------------------------------
```

[8]More information about the TSView database (7) used for DWOPSIM is found in appendix 1, which lists the country and product codes.

three regions sums to zero and that world supply balances world demand in each historical year.[9] The global-balancing criteria are found in the spreadsheet DWOPBLRW.CAL, which is used by DWOPTIME. Figure 2 shows what part of the USBF spreadsheet looks like after DWOPTIME is run. The spreadsheet output of DWOPTIME is formatted by the DWOPTEMP.CAL spreadsheet. This includes column titles (but not units) copied from the TSView database. A DOS batch program equivalent to DWOPTIME must be assembled for each new model. A convenient way of doing this is to invoke the program TIMEREPL (screen 1) with a new model name (with four letters or digits) and edit the new file in places that are bounded by "rem ***.." lines. For example, if a different model has five countries/regions instead of the three in TIME, extra lines will have to be added for the new countries.

DWOPTIME uses an ASCII file TIMERW.SUB, which also must be created for each new model. TIMERW.SUB is used by one of the TSView utility programs when RW data are derived by subtracting US and EC data from world (WD) data in the TSView database. This file, which defines the residual region RW for a model, must be properly created for each new model and saved on the model subdirectory. The full explanation of the format of TIMERW.SUB (fig. 3) is found in TSView documentation (2). Basically, this file causes TSView to subtract data for the

Figure 2--Part of USBF spreadsheet created by DWOPTIME

	A	B	C	D	E	F	G	H
1	USBF	Slaught.	Empty	Product.	Imports	Exports	Consump.	E.Stocks
2	Unit							
3	SSvVC	SL/AR	SW/YD	qPR	qIM	qEX	qCN	qES
4	Year							
5	1960	34644	.208	7111	352	4	7543	83
6		34551	.215	7426	470	15	7868	96
7		34768	.207	7195	352	16	7535	92
8		35274	.211	7426	470	16	7842	130
9		39310	.189	7411	653	15	8030	149
10	1965	40959	.219	8957	427	24	9387	122
11		41036	.228	9360	546	18	9866	144
12		40407	.236	9531	602	19	10128	130
13		41034	.239	9804	689	17	10469	137
14		40584	.244	9902	744	15	10603	165
15	1970	39559	.255	10103	824	18	10917	157
16		39730	.256	10184	797	24	10944	170
17		39335	.264	10374	905	28	11249	172
18		36506	.269	9808	917	41	10647	209
19		40528	.264	10715	747	29	11453	189
20	1975	46870	.240	11271	808	24	12080	164
21		48726	.250	12166	950	41	13024	215
22		48073	.246	11844	890	47	12754	148
23		44272	.255	11279	1053	74	12160	246
24		36932	.269	9924	1103	78	10982	213
25	1980	36795	.272	9998	946	80	10877	200
26		38151	.271	10354	799	100	11097	156
27		39264	.266	10425	889	115	11176	179
28		40136	.268	10748	873	125	11476	199
29		41259	.265	10929	838	152	11594	220
30	1985	40048	.275	10997	948	151	11819	195
31		41046	.275	11292	978	239	12036	190
32		38792	.281	10884	1040	277	11660	177
33		37889	.287	10880	1091	313	11641	194
34		36329	.293	10633	988	464	11197	154
35	1990	35245	.297	10465	1068	456	11048	183
36		35650	.297	10585	1030	472	11181	145
37		N/A	N/A	N/A	N/A	N/A	N/A	N/A

Figure 3--File TIMERW.SUB for model TIME

```
TEMP.TS                  <- Source file name  (NEWRW.SUB)
TEMP.TS                  <- Destination file name (can be same as source)
1                        <-- Number of new blocks to create
RW                        1  3  5
                   RW = WD-US-EC (Country codes for TS numbers above)
```

[9]TSView software contains a set of utilities that allows the user to manipulate TSView data and put it into spreadsheets. DWOPTIME and related programs make use of these TSView utilities.

US (TSView country #3) and the EC (country #5) from world data (TSView region #1). After DWOPTIME is run for a selected product, separate spreadsheets are left on drive D: for the product for each country designated in the DWOPTIME program. For example, when the command "DWOPTIME BF" is run for the product "beef and veal," the files USBF.CAL, ECBF.CAL and RWBF.CAL will be found on the D: drive. These files can then be edited and product-specific data can be added. The files must be manually copied to the TIME subdirectory for safekeeping and further use by the DWOPSIM modeling framework.

In addition to product/country files (such as USBF), country files (such as US) and a world reference price file (WD) with exogenous data also are required for the model TIME. Figures 4 and 5 show parts of these two files. Note in figure 5 that the current DWOPSIM spreadsheet format allows data to be simulated or projected up to the year 2020 (not all data are shown). Projects needing projections beyond this year would require an extensive reformatting of DWOPSIM template spreadsheets and some changes in DWOPSIM programs.

Figure 4--Spreadsheet US.CAL for the model TIME

	A	B	C	D
1	US	Time	Real In.	Popul.	Inc. Pc.
2		Unit	M.82US$	000's	82$/Per.
3		SSvVC	USeIN	USePP	USrIP
4		Year			
5		1960	1654100	180671	9155
6			1696600	183691	9236
7			1785600	186538	9572
8			1858500	189242	9821
9			1957100	191884	10199
10		1965	2070600	194303	10657
11			2192500	196560	11154
12			2255000	198712	11348
13			2347900	200706	11698
14			2406200	202677	11872
15		1970	2399100	205052	11700
16			2464100	207661	11866
17			2584900	209896	12315
18			2711800	211909	12797
19			2693500	213854	12595
20		1975	2665700	215973	12343
21			2793700	218035	12813
22			2921200	220239	13264
23			3073000	222585	13806
24			3136600	225055	13937
25		1980	3131700	227757	13750
26			3193600	230138	13877
27			3114800	232520	13396
28			3231200	234799	13762
29			3457500	237001	14589
30		1985	3581900	239279	14970
31			3687400	241625	15261
32			3820000	243934	15660
33			3988600	246329	16192
34			4087600	248777	16431
35		1990	4126200	250410	16478
36			4246272	252502	16817
37		.	4369839	254521	17169
38		.	4497001	256466	17534
39			4627864	258338	17914
40		1995	4762535	260138	18308
41			4901125	261764	18723
42		.	5043747	263389	19149
43		.	5190520	265015	19586
44			5341565	266640	20033
45		2000	5497004	268266	20491

Figure 5--Spreadsheet WD.CAL for the model TIME

	A	B	C	D	E
1	WD	Time	BF-R.P.	CN-R.P.	
2	Unit		89$/MT	89$/MT	
3	SSvVC	WDeTM	WDpBF	WDpCN	
4	Year				
5	1960	1	4858	203	
6		2	4460	193	
7		3	4558	193	
8		4	4198	211	
9		5	5213	210	
10	1965	6	5328	207	
11		7	5965	209	
12		8	5916	191	
13		9	5872	160	
14		10	6275	165	
15	1970	11	6337	175	
16		12	6186	166	
17		13	6502	152	
18		14	8281	249	
19		15	5980	310	
20	1975	16	4566	255	
21		17	5114	225	
22		18	4569	179	
23		19	6039	177	
24		20	7488	187	
25	1980	21	6571	185	
26		22	5370	176	
27		23	4875	137	
28		24	4793	165	
29		25	4299	159	
30	1985	26	3956	128	
31		27	3746	97	
32		28	4129	81	
33		29	4231	112	
34		30	4141	111	
35	1990	31	4105	109	
36		32	4077	107	
37	.	33	4048	104	
38	.	34	4020	102	
39		35	3992	100	
40	1995	36	3964	98	
41		37	3936	96	
42	.	38	3908	94	
43	.	39	3881	92	
44		40	3854	90	

Once the basic data for the spreadsheets have been prepared, the next step is to define the model structure by specifying equations and adding equation parameters. This is done in an equation specification spreadsheet, which accompanies every model spreadsheet containing variables to be modeled. The center of any DWOPSIM model becomes pairs of spreadsheets. In the case of the demonstration model TIME, there is a pair of spreadsheets for each product for each country. One spreadsheet--which is called the model spreadsheet--contains data and equations that forecast or project the data into the future. The other spreadsheet--which we will call the equation specification, or just equation spreadsheet--contains the coding of equations and equation parameters for the model spreadsheet.

Equation Parameters and Specification in Equation Spreadsheets

In a model spreadsheet, values of variables for years beyond the base year will require either equations calculating the variable or exogenous data values. Each DWOPSIM model spreadsheet holding equations MUST have an accompanying equation (specification) spreadsheet. This key equation information is used by DWOPSIM programs to write the equations in the appropriate place in the model spreadsheet. For example, equations in the model spreadsheet USBF.CAL will be written from the coded information in the accompanying equation spreadsheet USBFEQ.CAL. Figure 6 shows part of the equation specification spreadsheet USBFEQ accompanying the spreadsheet USBF.[10]

The equation spreadsheet holds equation terms in cells for every variable requiring an equation in a model spreadsheet. Explanatory variables for equations are found in columns C through AN. These explanatory variables may be contained in the model spreadsheet itself, or they may be contained in other model spreadsheets. Variables calculated by equations are represented by variable names in rows 6 through 43 of column A. Looking at figure 6, rows 1 and 2 contain descriptive variable names and units, respectively, which are used for reference purposes only. Row 3 contains the key model variable names that are used by DWOPSIM programs. The variable names have a specific format "SSvVC" where SS is the name of the spreadsheet containing the variable, v is

Figure 6--Part of the USBFEQ.CAL equation spreadsheet for the model TIME

	A	B	C	D	E	F	G	H
1	USBFEQ		SLaughter	Slaught.Wt.	PRoduction	IMports	EXports	CoNsumption
2	Base row>	36	number	KG/animal	1000 MT	1000 MT	1000 MT	1000 MT
3	Variable-(SSvVC)->		USBFnSL	USBFrSW	USBFqPR	USBFqIM	USBFqEX	USBFqCN
4	Variable column-->		B	C	D	E	F	G
5	.....							
6	BFnSL B				*v			
7	BFrSW C		(1/v)		*----> *v	<-v refers to		
8	BFqPR D				means	term in this		
9	BFqIM E	<--there is one row for			multiply	column (E)		*v
10	BFqEX F	each equation in the			*v			
11	BFqCN G	USBF spreadsheet						
12	BFqES H				*v			
13								

[10]Equation spreadsheet names take the name of the model spreadsheet appended with the suffix EQ. A listing of an equation spreadsheet for the model TIME can be found in appendix 2.

the general type of variable, and VC is a variable name code (see appendix 2 for a brief explanation of these codes). Row 4 contains the letter of the column in the model spreadsheet where the data for the variable resides. Cell B2 gives the row number in the model spreadsheet that contains the data to be used as the base year values for model initialization. The information in rows 3 and 4 is key to the operation of DWOPSIM model building programs.

Nomenclature conventions used in rows 3 and 4 of the equation spreadsheet must follow DWOPSIM conventions. The name of the model spreadsheet containing equations to be modeled (such as USBF) requires four letters. The name of the model spreadsheet MUST be located in cell A1 (right justified) in the equation spreadsheet and the appendix EQ MUST be (left justified) in cell B1. The name of the model spreadsheet containing a variable MUST be the first part of the variable name in row 3 of the equation spreadsheet. Model spreadsheets that have no equations (only exogenous data), must have a two-letter name (such as US).

An example can be given for BF production in column E of the equation spreadsheet USBFEQ.CAL (fig. 6). The variable is given the name USBFqPR, which means that the variable for the quantity (q) of beef production (PR) is found in column D of the model spreadsheet USBF (fig. 2). Column C, row 3 of USBFEQ.CAL in figure 6 gives the name USBFnSL for US beef slaughter numbers and row 4 says that this variable will be found in cclumn B of the spreadsheet USBF.CAL.

The cells below row 5 of the equation spreadsheet contain the equation specification and parameters for the variable term (column) as it appears in the equation (row). The equation for a variable is simply the concatenation of all of the column terms in an equation row (the name of the variable to be explained by the equation is found in column A of the equation row). If a cell is blank, as is the case with many cells in the figure 6, this means that the variable represented by the cell column does not appear in the equation for the variable equation in that row. If the cell contains information, it contains a "v," which means that the variable in that column is to be used as an equation term with whatever arithmetic is found in the cell. For example, in figure 5, cell E7 contains the code "*v." This means that the product (*) of this variable (US beef production = USqPR) is to used as a term in the equation calculating the slaughter weight of US beef (BFrSW). An examination of the entire row of an equation for a variable will illustrate the idea for a full equation, but that requires a listing of all the terms making up an equation (see appendix 2).

Because the equation spreadsheet is so important, a special output program DWOPEOUT is available to print out all of the information in the spreadsheet in full in a readable format. Screen 3 shows the screen seen when the DWOPEOUT program is invoked.[11] Note that if the P (print) option is chosen, the program calls a batch program HPLASER, which sets a small print on an HP laser printer. The software Laser Control is used for printing the small print in this report for the model TIME. A file PORTSMAL set in the software

---

[11]DWOPSIM program conventions follow those used in the SWOPSIM modeling framework. If a user is unsure of what a program does, typing the name of the program (without the model or variable names it asks for) will bring up the screen explaining the program. These screens, along with a full program listing for all programs, are found in appendix 3.

package defines the small print. If a user does not have this software, other laser printer control programs may do the job and their calling programs should replace HPLASER in DWOPSIM output programs. Alternatively, the user may select the F option, and print ASCII files to the D: drive, and use SIDEWAYS or some other printing routine to print the results in a readable format.

Screen 3--The program DWOPEOUT

```
                DWOPSIM Output Program
-------------------------------------------------------------------------
DWOPEOUT        Program to print two pages of a DWOPsim Equation and
                parameter spreadsheet OUTput FILE as well as a
                third page of model equations.
REQUIREMENTS    DWOPSIM equation spreadsheet file (FILEEQ.CAL)
                on NAME model subdirectory.  The template file DWOPEOUT.CAL
                must be available on the DWOPSIM subdirectory.  The print
                (P) option assumes an HPLASER batch program calling a
                PORTSMAL configuration file for the Laser Control program.
OUTPUT (D:)     Print two-page file on printer (P option) or on D: (with
                F option) and a FILEEOUT.CAL on D:.  Print one page of
                model variable names and equations (on printer or on D:).
-------------------------------------------------------------------------
COMMAND         DWOPEOUT  NAME FILE P (or F)
-------------------------------------------------------------------------
```

Figure 7 gives part of this full printout for row 8 of the file USBFEQ.CAL (see fig. 6), which contains the equation terms for the variable BFqPR, the quantity of beef produced in the US. The bottom of figure 7 shows the full equation in terms of the variable names (partially shown in row 3 of figure 6). This full equation for the US for BFqPR used a standard equation notation, which is based on SuperCalc 5 mathematical operations notation (* = multiply, / = divide, ^ = exponentiation). In effect, when the blank cells are removed, the equation is a constant plus a string of the remaining cells in the row. The equation at the bottom of figure 7 also illustrates how lagged variables are marked in the equation spreadsheet. The first three terms in the equation are producer incentive prices (PI) found in the commodity spreadsheet USBF for the US. The first price term is the current price, while the second and third are lagged prices where the period of lag is indicated by an appended 1 or 2, respectively, to the explanatory variable name. The last term in the equation is a growth term where 0.0116 is the supply growth rate and WDeTM is the time trend variable found in the world (WD) spreadsheet. The constant term in an equation does not appear in the equation spreadsheet. It is calculated and placed in the actual equation in the model spreadsheet when the spreadsheet is initialized to base period data.

The specification shown in figure 7 also shows linkages to other model spreadsheets, specifically the US country spreadsheet (fig. 4) that contains real income (USeIN) and population (USePP), which are required as explanatory variables. A linkage to the USCN model spreadsheet also occurs where the consumer price for corn (USCNpCN) is required as an explanatory variable with a demand elasticity of -0.05. Finally, a linkage to the world (WD) reference price spreadsheet containing real prices (fig. 5) and time trend (TM) variable occurs via the variable WDeTM.

From a practical point of view, the lined printouts of the equation spreadsheets, printed by DWOPEOUT (as shown in appendix 2 for the model TIME), give all the information about equations in model spreadsheets. Furthermore, these printouts can serve as worksheets if equations are to be modified because it is easy to read equation terms handwritten on the printouts and enter them into the equation specification spreadsheets. Control of equations

Figure 7--Row for variable BFqPR from the equation spreadsheet USBFEQ

	A	B	C	D	E	F
1	USBFEQ		SLaughter	Slaught.Wt.	PRoduction	IMports
2	Base row>	36	number	KG/animal	1000 MT	1000 MT
3	Variable-(SSvVC)->		USBFnSL	USBFrSW	USBFqPR	USBFqIM
4	Variable column-->		B	C	D	E
5	.....					
6	BFqPR D					

	G	H	I	J	K	L
1	EXports	CoNsumption	End.Stocks			
2	1000 MT	1000 MT	1000 MT			
3	USBFqEX	USBFqCN	USBFqES			
4	F	G	H			
5	.....					
6						

	M	N	O	P	Q	R
1						BF-PI.price
2						89.US$/MT
3						USBFpPI
4						S
5	.....					
6						*v^.1

	S	T	U	V	W	X
1	BF-PI.price	BF-PI.price	BF-CI.price	Net Trade	P.S.wedge	C.S.wedge
2	89.US$/MT	89.US$/MT	89.US$/MT	1000 MT	89.US$/MT	89.US$/MT
3	USBFpPI1	USBFpPI2	USBFpCI	USBFqNT	USBFwPS	USBFwCS
4	S	S	T	U	V	W
5	.....					
6	*v^.3	*v^.2				

	Y	Z	AA	AB	AC	AD
1	E.S.wedge	I.S.wedge	Prod.shFt.	Cons.shFt.		
2	89.US$/MT	89.US$/MT	Number	Number		
3	USBFwES	USBFwIS	USBFePF	USBFfCF		
4	X	Y	Z	AA		
5	.....					
6			*(1+v)			

	AE	AF	AG	AH	AI	AJ
1			CN-CN.price			
2			89.US$/MT			
3			USCNpCN			
4			T			
5	.....					
6			*v^-.05			

	AK	AL	AM	AN
1	WD Ref.P.	TiMe trend	real INcome	PoPulation
2	89.US$/MT	count	M.89.US$	1000s
3	WDpRP	WDeTM	USeIN	USePP
4	C	B	B	C
5	.....			
6	*(1+.0116)^v			

Variables and Equations in Spreadsheet--> USBF

Variable: USBFqPR
Variable description (unit), spreadsheet column: PRoduction (1000 MT), col. D

Var. Equation.....

USBFqPR = Constant*USBFpPI^.1*USBFpPI1^.3*USBFpPI2^.2*(1+USBFePF)*USCNpCN^-.05*(1+.0116)^WDeTM

is totally maintained by the information inserted into the equation specification spreadsheet. DWOPSIM programs transform the equation spreadsheet information into actual equations in the model spreadsheets. This will be discussed later, but we must discuss policy and exogenous data first.

Policy and Exogenous Data in Model Spreadsheets

A model spreadsheet contains historical and projected data. It is the repository for historical data for a model. With current formatting, historical data can begin from 1960. While historical data are useful for econometric analysis, the only historical data absolutely required for simulation modeling are data with enough years to account for lagged variables required in an equation. Then, there are the variables to be explained into the future. This may be done with equations relating particular variables to other variables in the host model or other model spreadsheets. Projected variables may also be entered exogenously, or they may be calculated by equations (identities) manually entered into the spreadsheet. But, projected numbers are required for (exogenous) variables that are used in simulation equations; these numbers often are policy variables or exogenous data, from a modeling viewpoint. All of these data, no matter how they are generated, can be printed out by the program DWOPBOUT (screen 4). Again, if an HP laser printer is attached and the appropriate INSIGHT software is used, DWOPBOUT will print out a page of condensed print. If the "F" print to disk option is invoked, the user can obtain a print copy by some other means.

Figure 8 shows part of the data printout from the program DWOPBOUT for the model spreadsheet USBF from the DWOPSIM demonstration model TIME. Several points concerning prices, net trade, and policy price wedges can be illustrated with the data shown. Net trade is a variable generated with an "in spreadsheet" formula for both historical and future periods; net trade is an identity and does not require an explanatory equation with an intercept. This illustrates the point that formulas for variables not requiring constant terms can be put into spreadsheets manually. The other data in figure 8 do not exist in the model prior to 1989. But as long as simulations into the future for any variable do not require these variables (lagged) prior to 1989, simulation will be possible. Finally, some of the data are "policy" data, which are used to emulate policy changes over time. In the case of BFwCS, a consumer subsidy wedge, the numbers decline over time and are simply a base period number projected into the future with a deflation formula.

The flexibility of the spreadsheet enters into full force in these model spreadsheets. Spreadsheet formulas can be used to generate data if that is a desirable way to go. Data can be prepared externally to the model spreadsheets and can be entered manually or by user-designed programs.

In the case of identities that do not require constant terms, two options are available. Formulas can be entered manually into model spreadsheets, or they can be coded as equations in equation spreadsheets to be written into the model spreadsheets

Screen 4--The program DWOPBOUT

```
                DWOPSIM Output Program
-------------------------------------------------------------------------
DWOPBOUT        Program to print a page of a DWOPsim Base data spreadsheet
                OUTput.
REQUIREMENTS    DWOPSIM model spreadsheet with assumed FILE name (e.g.
                USBF.CAL) on NAME model subdirectory.  The template file
                DWOPBOUT.CAL must be available on the SWOPSIM subdirectory.
                The print (P) option assumes an HPLASER batch program
                calling a PORTSM97 configuration file.
OUTPUT (D:)     Print 1 page file on D: (with P option) or on D: (with F
                option) and a FILEBOUT.CAL on D:
-------------------------------------------------------------------------
COMMAND         DWOPBOUT  NAME FILE P (or F)
-------------------------------------------------------------------------
```

Figure 8--Part of the model spreadsheet USBF

Col.>	P	Q	R	S	T	U	V	W	X
USBF				BF-PI.pr	BF-CI.pr	Net Trad	P.S.wedg	C.S.wedg	E.S.wedg
Unit				89.US$/M	89.US$/M	1000 MT	89.US$/M	89.US$/M	89.US$/M
SSvVC				BFpPI	BFpCI	BFqNT	BFwPS	BFwCS	BFwES
Year									
1960	.	.	.	.00	.00	-348.00	.00	.00	.00
	.	.	.	.00	.00	-455.00	.00	.00	.00
	.	.	.	.00	.00	-336.00	.00	.00	.00
	.	.	.	.00	.00	-454.00	.00	.00	.00
	.	.	.	.00	.00	-638.00	.00	.00	.00
1965	.	.	.	.00	.00	-403.00	.00	.00	.00
	.	.	.	.00	.00	-528.00	.00	.00	.00
	.	.	.	.00	.00	-583.00	.00	.00	.00
	.	.	.	.00	.00	-672.00	.00	.00	.00
	.	.	.	.00	.00	-729.00	.00	.00	.00
1970	.	.	.	.00	.00	-806.00	.00	.00	.00
	.	.	.	.00	.00	-773.00	.00	.00	.00
	.	.	.	.00	.00	-877.00	.00	.00	.00
	.	.	.	.00	.00	-876.00	.00	.00	.00
	.	.	.	.00	.00	-718.00	.00	.00	.00
1975	.	.	.	.00	.00	-784.00	.00	.00	.00
	.	.	.	.00	.00	-909.00	.00	.00	.00
	.	.	.	.00	.00	-843.00	.00	.00	.00
	.	.	.	.00	.00	-979.00	.00	.00	.00
	.	.	.	.00	.00	-1025.00	.00	.00	.00
1980	.	.	.	.00	.00	-866.00	.00	.00	.00
	.	.	.	.00	.00	-699.00	.00	.00	.00
	.	.	.	.00	.00	-774.00	.00	.00	.00
	.	.	.	.00	.00	-748.00	.00	.00	.00
	.	.	.	.00	.00	-686.00	.00	.00	.00
1985	.	.	.	.00	.00	-797.00	.00	.00	.00
	.	.	.	.00	.00	-739.00	.00	.00	.00
	.	.	.	.00	.00	-763.00	.00	.00	.00
	.	.	.	.00	.00	-778.00	.00	.00	.00
	.	.	.	2636.00	4779.00	-524.00	.00	14.00	.00
1990	.	.	.	2559.22	4639.81	-612.00	.00	13.59	.00
	.	.	.	2555.25	4575.23	-558.00	.00	13.20	.00
.	.	.	.	2525.54	4545.31	-720.69	.00	12.81	.00
.	.	.	.	2496.07	4516.81	-816.49	.00	12.44	.00
	.	.	.	2466.83	4487.93	-932.70	.00	12.08	.00
1995	.	.	.	2437.82	4459.28	-1048.24	.00	11.72	.00

when they are initialized by the program DWOPINIT.

Once model spreadsheets contain historical data and any generated variables required for simulation past a model base period, the user is ready to initialize model spreadsheets. All of the required DWOPSIM operations for data preparation could be done manually. Alternatively, if an appropriate TSView supply and utilization database is available, DWOPSIM programs could automate some of the more difficult parts of data preparation. Exogenous projected data and derived variables will have to be entered manually.

Once data entry is completed, programs such as DWOPEOUT and DWOPBOUT[12] are available to print out compact readable versions of the model spreadsheets.

---

[12]DWOPEOUT also prints a page listing model spreadsheet variables and the equations contained in the spreadsheet using variable names found at the top of the equation spreadsheet. These "equations" (identical to the one seen at the bottom of figure 7 and listed in full for the model TIME in an appendix) serve as the mathematical documentation of a model.

Initialization of a Model Spreadsheet

Given a coding of equations structure and parameters in the equation spreadsheet, the next step is to have equations generated for years beyond the base period in the model spreadsheet. This converts the model spreadsheet from a data repository into a spreadsheet with equations explaining selected variables as a function of variables in the host and other model spreadsheets.

The program DWOPINIT writes equations and calculates intercepts (constant terms) that make the equations replicate base year data. Screen 5 shows the program DWOPINIT. Once the program has been run for a model (country and commodity, in case of the demonstration model TIME), it leaves the spreadsheet operated upon as well as spreadsheets linked to it by equations on the D: drive. These spreadsheets must be manually copied to the model subdirectory for permanent safekeeping.

The equations are written into the model spreadsheet in simple formula, using spreadsheet cell notation to refer to variables. The bottom line of figure 9 shows the formula appearing in cell D36 of the spreadsheet USBF that was written out by the DWOPINIT program. The first thing to notice about the equation is that variables are now cell references in the USBF spreadsheet or cell references in other spreadsheets (the cell reference then is preceded by the spreadsheet name and a "!" mark). For example, the term USCN!T36 refers to the variable in cell T36 of the spreadsheet USCN. The cross spreadsheet linkage capability of SuperCalc 5 allows equations to include terms that are linkages to other spreadsheets. Also notice that the constant term is a number. This is calculated by the DWOPINIT program from the equation itself and the base period data (recall that the base period is selected in USBFEQ.CAL by marking the base period row in cell B2) and is inserted at the beginning of the equation. The arithmetic in the equation uses the notation of formulas in the SuperCalc 5 spreadsheet.

Screen 5--Program DWOPINIT for initialization of a model spreadsheet

DWOPSIM Program

DWOPINIT	Program to INITialize equations in a model spreadsheet, given the equation cell components. Equations with a calculated constant term are inserted from the base period forward. A FILE name is assumed for the model spreadsheets (e.g. USBF.CAL).
REQUIREMENTS	A C:\NAME subdirectory must contain the model spreadsheet FILE and the FILEEQ equation specification spreadsheet. The equation writing spreadsheet must exist on the C:\DWOPSIM subdirectory.
OUTPUT (D:)	The initialized spreadsheet FILE on D:.
COMMAND	DWOPINIT NAME FILE

Figure 9--Equation for BFqPR in model spreadsheet USBF

	A	B	C	D	E	F	G	H
34		36329	.293	10633	988	464	11197	154
35	1990	35245	.297	10465	1068	456	11048	183
36		35650	.297	10585	1030	472	11181	145
37	.	35831	.297	10639	1046	474	11359	146
38	.	36107	.297	10721	1063	478	11536	147

USBF!D36

82.52997159*S36^.1*S35^.3*S34^.2*(1+Z36)*USCN!T36^-.05*(1+.0116)^WD!B36

Every time a data number or equation structure or parameter is changed in a model spreadsheet for a country, DWOPINIT must be run to update the equations and recalculate the constant term. If there are many model spreadsheets for many products and countries, a batch program that loops through all the model spreadsheets containing equations calling DWOPINIT should be prepared. The demonstration program TIME has a batch program TIMELOOP that does this kind of looping operation.

Some model spreadsheets may contain only exogenous data and no equations need to be written. In this case, there will be no accompanying equation spreadsheet nor any need to run DWOPINIT. The country spreadsheets US, EC, and RW in the model TIME are examples of these "equation-less" spreadsheets.

Assembly of Model Spreadsheets Into a World Model Linked in Memory

When the model spreadsheets (such as product, country, and world spreadsheets in the demonstration model TIME) have all been prepared and initialized, the user runs the DWOPSIM program DWOPSOLV. This program, shown in screen 6, calls up the equation parts (as well as the required exogenous data in the model spreadsheets) and places them in computer memory. DWOPSOLV also gets the world reference price spreadsheet, adds a world market-clearing mechanism template, and places this new world spreadsheet into memory. This is done in a manner that maintains linkages across spreadsheets. When DWOPSOLV has completed model assembly, the screen will show a message telling the user to press "Enter" to gain control of the world model. At this point, the model is ready to solve. When "Enter" is pressed, the user will observe a split screen centered on parts of the world market solution and reference price spreadsheet. The cursor can be moved to observe different parts of the spreadsheet on the screen. Entering ";" will toggle the cursor between the two screen parts. The screen size has been set for zoom, which uses small print to give a larger picture of action on the screen. The computer commands (CNTL+) or (CNTL-) will toggle through the spreadsheets held in memory. Note that when DWOPINIT is invoked, the user must enter a one-digit/letter code to mark the solution when it is saved with the ALT-S command. This gives the user markings for up to 36 solutions for one assembled world model.

Screen 6--Program DWOPSOLV to assemble a world model from individual model spreadsheets

DWOPSIM World Model SOLVer Assembly Program	
DWOPSOLV	Program to assemble a DWOPSIM world model in memory and prepare it for SOLVing. A macro is also created which can be used to save the results on D: after the model is solved. Type ALT-S to invoke the 'save' macro. The model can be solved again with the program NAMEAGAN.
REQUIREMENTS	DWOPSIM model spreadsheets in the NAME subdirectory. A 1 Digit letter or number must be selected to mark the solution value files saved on D: This program also requires that a model specific batch file has to have been created and saved on the batch subdirectory (under the name NAMESOLV.BAT). This file tells DWOPSOLV what model files comprise the global model to be assembled. A WDS.CAL world solution mechanism file must have been created on the NAME subdirectory.
OUTPUT (D:)	Linked files in memory for the model NAME, ready for SOLVing. If the ALT-S macro is invoked after the model is solved, D: will contain solution value files appended with the selected Digit (e.g. if D=1 for USBF, file is USBF.S1).
COMMAND	DWOPSOLV NAME D

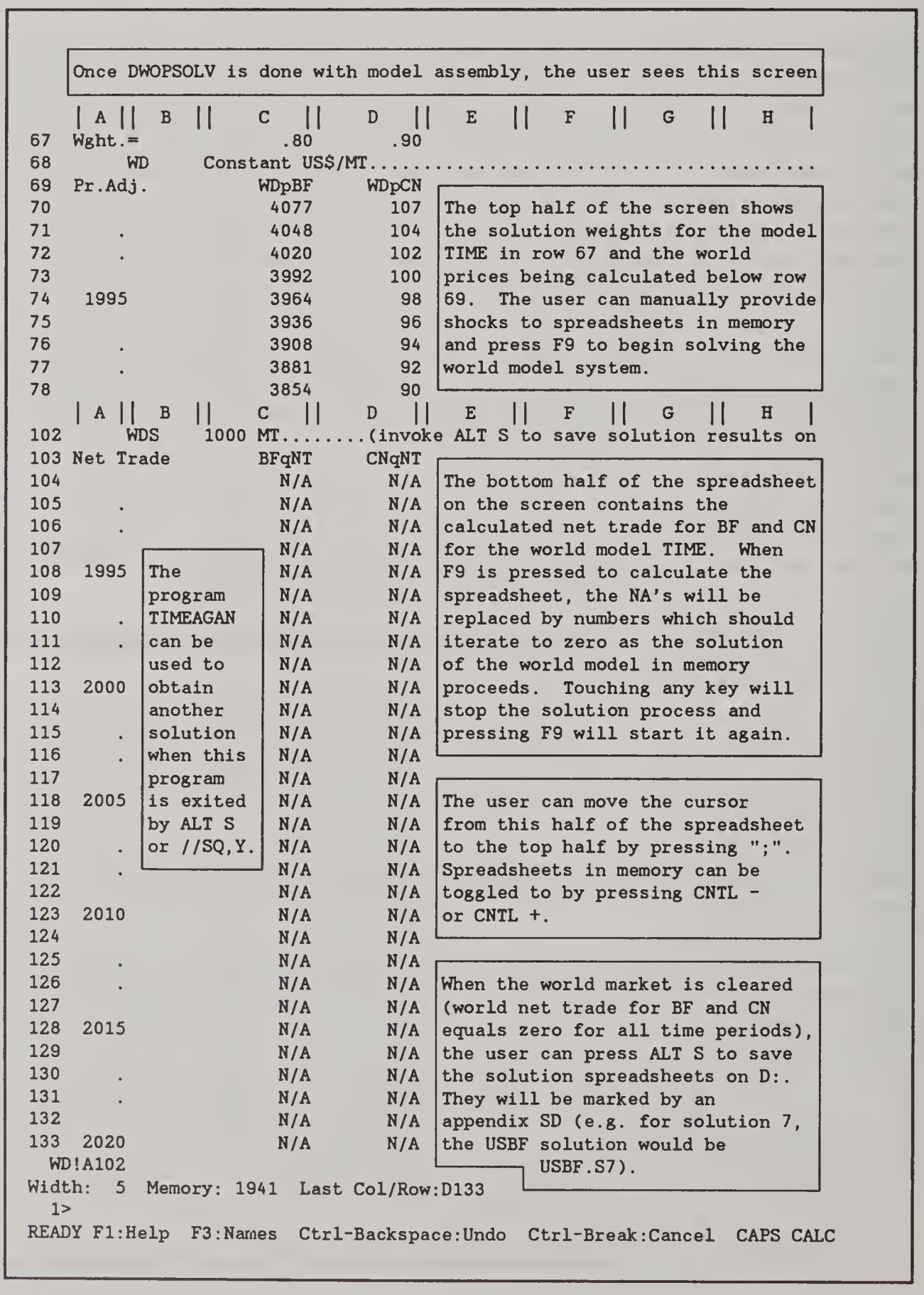
Simulation of a DWOPSIM World Model and the Reporting of Simulation Results

The top part of figure 10 shows the message telling the user to proceed, and the bottom part shows the screen the user sees when the world model is ready to be solved. At this point, the user may manually change exogenous or policy data to shock the model. This may be done, for example, by manually changing projected (baseline) values of policy variables or by changing intercept shift terms if they have been entered as equation variables. When preparation is done in the appropriate model spreadsheets in memory, the global system can be solved.

Once the recalculation command is issued (by pressing F9), the set of spreadsheets in memory will solve simultaneously until world markets are cleared by the world market-clearing mechanism now added to the WD.CAL spreadsheet in memory. This solution mechanism has to be manually constructed prior to solution and saved in the model subdirectory as a WDS.CAL file. In the case of the demonstration model TIME, this mechanism changes world prices for BF and CN in all time periods following the base period until world trade equals zero for BF and CN in all time periods.

Once the model has begun solving, the

Figure 10--Screens showing the world model is ready to solve



user can let it complete solution by itself or he may want to visually check net world trade figures for progress in moving toward zero. The solution mechanism shown for the demonstration model TIME does have damping factors that can be adjusted for each product to control the world reference price and net trade swings (see fig. 10). These parameters are found in the world market-clearing template with the name WDS.CAL associated with each model. The market-clearing mechanism for the model TIME itself is similar to that of SWOPSIM where world prices are changed depending upon world net trade balances. When world net trade is zero for all products, world prices do not change anymore and the model is solved.

When an acceptable solution has been reached, the user must type in "ALT-S" to execute a macro that saves the solution values for each component spreadsheet of the world model on the D: drive. This solution can be viewed at will and should be saved to the model subdirectory for safekeeping and later analysis. The solution spreadsheets are labeled with the spreadsheet name with an appendix, which is S followed by a number or digit marking the solution. This number or digit was entered when the program DWOPSOLV was invoked. Once the solution has been saved in memory, programs are available to examine the output.[13]

The program DWOPSOLV itself calls a model specific solution program that must be prepared for each model. In the case of the model TIME, this program on the batch subdirectory has the name TIMESOLV. As was the case with DWOPTIME, the program TIMESOLV can be edited to suit the coverage of a different DWOPSIM model. See appendix 3 for a listing of DWOPSOLV. DWOPSOLV also calls a program DWOPWDS, which adds the world market-clearing mechanism in the spreadsheet WDS.CAL to the WD.CAL world spreadsheet in memory.

If the user does not want to save the simulation, the normal SuperCalc 5 commands //SQ,Y will exit the model spreadsheets in memory. When either the solution has been saved on D: (by the ALT S command) or the world model has been exited (via the //SQ,Y SuperCalc 5 sequence), the user has the option of obtaining OTHER solutions by running a model specific *AGAN program. In the case of the model TIME, this program is TIMEAGAN. It takes the model files that DWOPSOLV had prepared for

Screen 7--The program DWOPCVAR

```
                DWOPSIM Output Program
-------------------------------------------------------------------------
DWOPCVAR        Program to compare up to four solution VALues with baseline
                and historical values for a selected spreadsheet variable
                (Column) in a model spreadsheet FILE (e.g. USBF).
REQUIREMENTS    DWOPSIM spreadsheet and solution spreadsheet(s) on NAME
                model subdirectory.  The template file DWOPCVAR.CAL
                must be available on the DWOPSIM subdirectory.
                If the printout is invoked from the output spreadsheet,
                an HPLASER batch program calling a PORTSMAL configuration
                file for the Laser Control program is assumed available.
OUTPUT (D:)     Comparisons on screen (spreadsheet with graph in it)
-------------------------------------------------------------------------
COMMAND         DWOPCVAR  NAME FILE C S1 S2 S3 S4
-------------------------------------------------------------------------
```

[13]Imagination in examining, manipulating, and presenting output results is largely up to the model builder. Spreadsheets are a very good medium for this purpose because spreadsheet templates, which reach into solution and history/baseline spreadsheets, can be constructed to aid in this process.

simulation and saved on D: and re-installs them into memory for another solution scenario. This process can be continued indefinitely, depending on room for storage of the solution values.

Once simulation results have been saved, three programs are available to examine and report simulation results. Screen 7 shows the screen for the program DWOPCVAR, which compares variables for up to four solutions with historical/baseline data.

Part of the spreadsheet created by DWOPCVAR is shown in figure 11. This spreadsheet can be printed out by executing the macro shown at the bottom of the figure (use ALT-F5). The template spreadsheet DWOPCVAR.CAL on the C:\DWOPSIM subdirectory, formats the output data and graphics. Figure 12 shows the graphical output contained in the spreadsheet created by DWOPCVAR. Again, the format and contents of the graph are created in the template spreadsheet DWOPCVAR.CAL. This illustrates the general method of reporting results, which is to create small programs and templates to retrieve and create the desired output. Because the solutions and historic or baseline data are in spreadsheets, all forms of reports and graphics can easily be created by the user. This can include the conversion of SuperCalc 5 spreadsheet output into formats readable by other spreadsheets.[14]

In figure 12, the solution files have been marked by the digit "7" and the plot shows the projection of US beef production for history (1960-91) and the base projection and alternative solution "7" from 1992 to 2020. The C:\TIME subdirectory has a series of commodity files with labels USBF7, USCN7, etc., which contain the solution data. The "7" solution shows the value of US beef production when world markets have cleared, while the baseline is simply the trend value of production. Normal procedure would be to substitute a baseline figure created when world markets cleared and then do alternative scenarios from the "globally balanced" baseline.

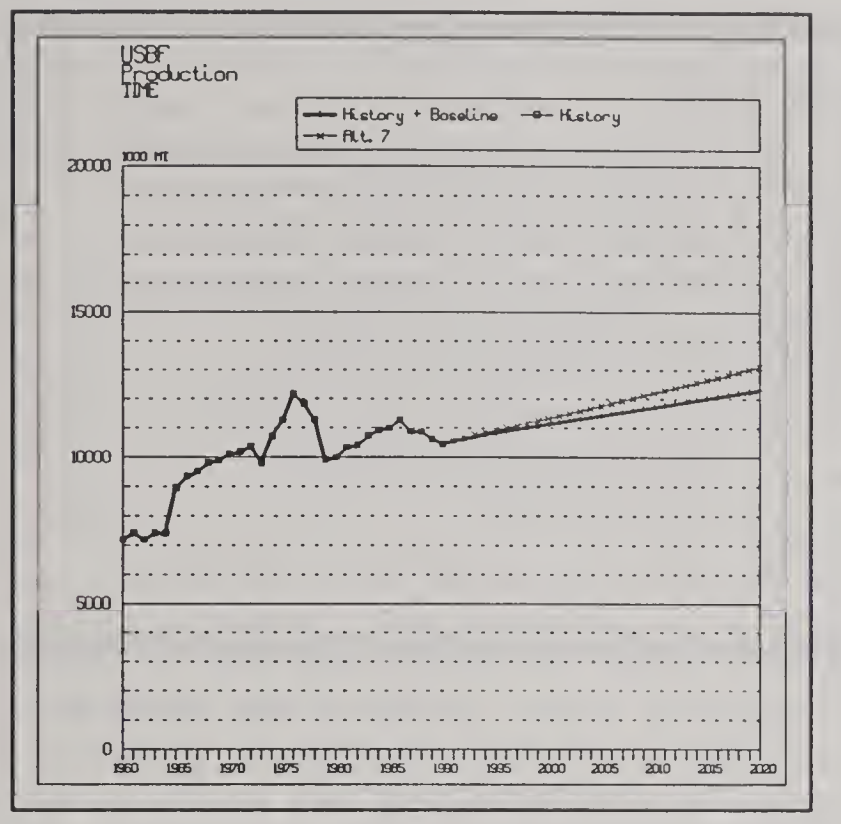
Solution output can be printed in the same format as historical and baseline data with the program DWOPSOUT. The output is in exactly the same format as that produced by DWOPBOUT (see output of DWOPBOUT in fig. 8 and app. 2). Screen 8 shows the screen seen if DWOPSOUT program is executed without

Figure 11--Part of an output spreadsheet created by DWOPCVAR

	A	B	C	D	E	F	G	H
1	DWOPCVAR							
2		BASE	.....Alternative Scenarios.....					
3	USBF	PRoducti	7				7	
4	Unit	1000 MT	<----------				.....Differen	
5	SSvVC	BFqPR	<----------				1000 MT	<----
6	Year	.....	.....	.....	.....	.....	.....	.....
7	1960	7195	N/A	N/A	N/A	N/A	N/A	
8		7426	N/A	N/A	N/A	N/A	N/A	
9		7195	N/A	N/A	N/A	N/A	N/A	
...								
34		10884	N/A	N/A	N/A	N/A	N/A	
35		10880	N/A	N/A	N/A	N/A	N/A	
36		10633	N/A	N/A	N/A	N/A	N/A	
37	1990	10465	N/A	N/A	N/A	N/A	N/A	
38		10585.00	10585.00	N/A	N/A	N/A	-4.58e-7	
39		10638.88	10660.64	N/A	N/A	N/A	21.75653	
40		10720.67	10771.89	N/A	N/A	N/A	51.22141	
41		10780.70	10859.36	N/A	N/A	N/A	78.65336	
42	1995	10841.06	10934.78	N/A	N/A	N/A	93.71807	
43		10901.76	11012.84	N/A	N/A	N/A	111.0758	
44		10962.12	11091.94	N/A	N/A	N/A	129.8193	

[14]SuperCalc can output LOTUS spreadsheets as an option. LOTUS spreadsheets are readable by most spreadsheet programs and many graphics presentation packages such as Harvard Graphics.

Figure 12--Graphical output from DWOPCVAR



providing parameters for it. The printed data series for both DWOPSOUT and DWOPBOUT go from 1960 to 2005 in order to get everything on one page with reduced print. Of course, this option could be changed by changing parameters in the programs and relevant spreadsheet templates.

Another useful output routine, DWOPCSOL, is similar to DWOPCVAR. DWOPCSOL produces a spreadsheet and graph of up to four variables from a model solution spreadsheet. It is useful for looking at graphs of a set of variables, compared with history and baseline in a model spreadsheet. Screen 9 shows the screen seen when DWOPCSOL is invoked without any of the required parameters.

If DWOPSIM proves to be a practical vehicle for baseline work, projections, and time series modeling, users will use the simple techniques of DWOPCVAR, DWOPSOUT, and DWOPCSOL to develop other output routines. All of these programs are simply DOS batch programs, which manipulate model spreadsheets and spreadsheet templates that format the output.

Concluding Comments

The building blocks of DWOPSIM are simple compared with those of SWOPSIM (6). This is because the large issue of model specification for the DWOPSIM model builder is not built into the DWOPSIM framework itself as was the case with SWOPSIM. This means that there are fewer programs and procedures to deal with, but it also means that the DWOPSIM user will have to cope with the unique and complex problems that come with building a solvable and well-behaved time series model. Many of the DWOPSIM conventions were derived from

Screen 8--The program DWOPSOUT

```
                DWOPSIM Output Program
-------------------------------------------------------------------------
DWOPSOUT        Program to print a page of a DWOPsim Solution from a model
                spreadsheet OUTput FILE.
REQUIREMENTS    DWOPSIM model spreadsheet solution FILE on model
                subdirectory (NAME).  The solution spreadsheet will have
                the suffix *.SD where D is 1 Digit solution marker.
                A spread model name (FILE) is assumed.  The template file
                DWOPSOUT.CAL must be available on the SWOPSIM subdirectory.
                The print (P) option assumes an HPLASER batch program
                calling a PORTSM97 configuration file for the Laser Control
                program.
OUTPUT (D:)     Print 1 page file on D: (with P option) or on D: (with F
                option) and a FILESD.CAL on D:
-------------------------------------------------------------------------
COMMAND         DWOPSOUT  NAME FILE D P (or F)
-------------------------------------------------------------------------
```

SWOPSIM modeling experience, and these have evolved because of the need to keep spreadsheet base models coded as efficiently as possible.

Screen 9--The program DWOPCSOL

```
                    DWOPSIM Output Program
-------------------------------------------------------------------------
DWOPCSOL        Program to Compare up to 4 variables in a SOLution with
                historical/baseline values.  Variables (Columns) are in
                a model FILE (e.g. USBF) and solution values are in a
                solution file (e.g. USBF.SD) where D marks the solution.
REQUIREMENTS    DWOPSIM model spreadsheet and solution spreadsheet(s) on
                model subdirectory.  The template file DWOPCSOL.CAL
                must be available on the DWOPSIM subdirectory.
                Printing (an option from the comparison spreadsheet)
                assumes an HPLASER batch program calling a PORTSMAL
                configuration file for the Laser Control program.
OUTPUT (D:)     Comparisons on screen (spreadsheet with graph in it)
-------------------------------------------------------------------------
COMMAND         DWOPCSOL  NAME FILE D C1 C2 C3 C4
-------------------------------------------------------------------------
```

If a DWOPSIM user can begin work with an existing TSView supply and utilization database, which has been formulated to use two-digit country and commodity codes (such as the SWOPSIM 1989 TSView database (7)), DWOPSIM programs are available to prepare globally balanced sets of supply and utilization data for products defined in the TSView database. This process creates DWOPSIM product spreadsheets similar to the TSView database format.

Other data (not from TSView) needed for modeling in a world model must be found and entered manually into appropriate spreadsheets. A world market-clearing mechanism must be created as well. The user is free to design the spreadsheet structure of a model, or the user can proceed by copying the scheme that is used in the demonstration model TIME.

The user chooses equation specification and must provide associated parameters for econometric work or other sources. Equation specification can be linear or nonlinear, and variables can be lagged up to 9 years. Equations in a model spreadsheet can refer to other variables in that spreadsheet or other spreadsheets. For each model spreadsheet requiring equations, the user must code the selected equation specification and parameters into the accompanying equation spreadsheet. When this is done, programs initialize the model spreadsheets by writing the equations and including their calculated constant terms, which make the equations replicate base period data.

Once the model spreadsheets are initialized, the DWOPSOLV program assembles all of them in memory to create a linked global model. The user then solves them simultaneously and saves the solution. Subsequent scenarios can be obtained by resolving the world model files that were prepared by DWOPSOLV. Programs are available to examine solution output from various perspectives.

The simple structure of DWOPSIM models and the fact that they are built and run in spreadsheets make it relatively easy for users to prepare their own methods of examining output and doing post-solution analysis.

Key information on model structure and the location of variables is found in the equation spreadsheets. The choice of where to locate variables (which columns of a spreadsheet) is largely up to the model builder. However, it is wise to be consistent across spreadsheets in order to simplify the preparation of the three DWOPSIM programs that must be customized for each model. Consistency is needed to simplify the preparation of the world market-clearing

mechanism and to take advantage of programs that can manipulate TSView data.[15]

When the user is familiar with DWOPSIM, the printouts of data and equation specification and parameters by the programs DWOPBOUT and DWOPEOUT, will provide adequate model recordkeeping.

The model TIME used to illustrate DWOPSIM is a three-region, two-product world model that includes features showing the capability of DWOPSIM. These features include cross-commodity (and cross-spreadsheet) equation linkages, lagged variables in equations, and various equation forms, including both linear and nonlinear specifications. The user can view the equations in the TIME model equation spreadsheets.

DWOPSIM equations are efficiently written using spreadsheet cell references, but for exposition purposes they are printed out by the program DWOPEOUT using variable names. The creation of a DWOPSIM model with more spreadsheets than found in the demonstration model TIME involves the same principles illustrated with this small model.

Installation of the DWOPSIM framework is a simple matter. The disk containing the programs includes instructions and an installation program. The computer containing DWOPSIM has to be configured with a D: drive (use a virtual disk or the DOS ASSIGN command) and should have 4 MB of memory with software to make it usable by SuperCalc 5. The installation program will create the proper subdirectories. If the user has an HP laser printer, he should also install the HPLASER program on the disk according to the instructions. This will make it possible to print DWOPSIM output in small print.

Finally, experience with time series model building with DWOPSIM will generate new and better ways of doing things. Further developments might include the automatic, as opposed to manual, creation of model-specific programs, such as TIMESOLV. However, future steps such as this will depend upon experience gained with DWOPSIM. In model building, programming cannot get ahead of actual user needs. All users are encouraged to record their experiences with DWOPSIM and provide suggestions for improvement. In the end, time series modeling itself is the best teacher.

---

[15]In the case of the model TIME, the three programs were (1) DWOPTIME which was needed to assemble commodity spreadsheet data from TSView data, (2) TIMESOLV which was needed by DWOPSOLV in the world model assembly stage, and (3) TIMEAGAN which can be used for repeated solution scenarios once DWOPSOLV has been run.

References

1. Dixit, Praveen, and Vernon Roningen. Modeling Bilateral Trade Flows with the Static World Policy Simulation (SWOPSIM) Modeling Framework. Staff Report AGES861124, Econ. Res. Serv., U.S. Dept. Agr., Dec. 1986.

2. Gudmunds, Karl, Alan Webb, and Gene Hasha. "TS Software, User Manual for the TIME SERIES Program and Utilities." Econ. Res. Serv., U.S. Dept. Agr., forthcoming.

3. International Monetary Fund. International Financial Statistics, various issues.

4. Labys, Walter. Dynamic Commodity Models: Specification, Estimation, and Simulation. Lexington Books, D.C. Heath and Company, Lexington, MA, 1973.

5. Roningen, Vernon, Praveen Dixit, and Ralph Seeley. "Agricultural Outlook in the Year 2000: Some Alternatives," Agriculture and Governments in an Interdependent World. Allan Maunder and Alberto Valdes (eds.), Dartmouth Pub. Co., London, England, 1990.

6. Roningen, Vernon, John Sullivan, and Praveen Dixit. Documentation of the Static World Policy Simulation (SWOPSIM) Modeling Framework. Staff Report No. AGES 9151, Econ. Res. Serv., U.S. Dept. Agr., Sept. 1991.

7. Sullivan, John, Vernon Roningen, Susan Leetmaa, and Denice Gray. A 1989 Global Database for the Static World Policy Simulation (SWOPSIM) Modeling Framework. Staff Report No. AGES 9215, Econ. Res. Serv., U.S. Dept. Agr., May 1992.

8. U.S. Department of Agriculture, Economic Research Service. "Estimates of Producer and Consumer Subsidy Equivalents: Government Intervention in Agriculture, 1982-1989." Forthcoming.

9. U.S. Department of Agriculture, Foreign Agricultural Service. "Production, Supply, and Distribution Database" [computer file]. Unpublished, Nov. 1991.

Appendix 1--Overview of a TSView Database Used by DWOPSIM

The TSView system (2,7) is a database and file viewer system developed by the Economic Research Service (ERS) to house supply and utilization data for grains, oilseeds, and livestock products. TSView also allows the user to select data from a database and put it out in a form that is readable by spreadsheets. This and other capabilities of TSView can be automated with utility programs that come with the system. DWOPSIM DOS batch programs combine these utilities with spreadsheet macros and templates to automate the process of preparation of a globally balanced dataset for a set of countries in a model for a selected commodity. The DWOPSIM programs require that the TSView database use two-letter country and commodity codes and that the programs themselves must be keyed to the numerical position of a particular country in the TSView database. Then balanced supply and utilization data for world models with any mix of countries and commodities can be easily created.

The TSView database behind this report can be referred to as the **DWOPSIM/SWOPSIM (DS)** TSView database. The key codes for DWOPSIM users are the two-letter code for each country/region and two-letter product code for each commodity (or aggregate). These codes are used not only to access the DS database, but they are also used in DWOPSIM spreadsheet names created from the data and, therefore, in variable names themselves. If the user wishes to use different codes and still use the TSView database, the choice is to re-do the database or manually rename everything in model spreadsheets (including spreadsheet names) derived from the DS database.

This appendix lists the numerical and alphabetical codes for all the countries/regions in the FAS TSView database used by the Commodity Economics Division, ERS, and in the DS version of the TSView database that contains additional aggregate regions for the different global databases. As a result, there are 213 countries/regions in the intermediate version of the CED TSView database, from which we have selected the following list of 143 countries/regions for use in the DS global database.

A detailed listing of the countries/regions within each of the 25 additional aggregate regions (213 minus 189) is provided by numerical codes here and by name in the appendix for each global database where they appear. The rest-of-world region for each database is provided by subtraction. This illustrates that the rest-of-world region differs by database, but does not list exactly what countries/regions it contains, since it is also used as a trade-balancing region. For some SWOPSIM databases, the rest-of-world region is quite large.

For reference, a listing in numerical order, and by name, of the 189 countries/regions contained within the FAS PS&D database follows. Then commodities included in the DS database are listed. There are 47 commodities listed by name and two-letter alphabetical code, which are contained in the DS version of the TSView database. The first set listed is the standard 22-commodity set used in most ERS SWOPSIM models, and includes four commodities that are calculated aggregates. Next are the seven additional commodities that are now routinely included in the SWOPSIM database "t" files, and are available in a form ready for incorporation into models. Finally, we list the 18 additional commodities available in the DS TSView database. Of these, 12 come from the 4 aggregate commodities in the standard 22-product (SWOPSIM) set, and 4 come from the 4 new aggregates that result when the 12 are removed from the original aggregates. The last two commodities result from the poultry meat being separated into turkey meat, and broiler and other meat.

Concordance Between the FAS 189 Country/Region TSView Database and DWOPSIM/SWOPSIM Global Databases

Master Concordance Between FAS PS&D Database and DWOPSIM/SWOPSIM TSView Global Model Databases

Number in FAS 189 Country PS&D Database	DWOPSIM/SWOPSIM TSView Database 2 Letter Code and Country/Region Name	TSView ORDER	SWOPSIM 2 Digit Global Model Code for Model Containing Country/Region WD = WorlD, EU = EUrope, WH = Western Hemisphere, AS = ASia and Pacific Rim, AF = AFrica
1	WD - WorlD	1	(Aggregate)
2	FG - ForeiGn (non-US)	2	(Aggregate)
6	US - United States	3	WD EU WH AS AF
7	CN - Canada	4	WD EU WH AS
44	EC - European Community	5	WD WH AS AF
58	WE - O.W. Europe	6	WD AS
156	JP - Japan	7	WD EU WH AS
180	AU - Australia	8	WD EU AS
181	NZ - New Zealand	9	WD EU AS
151	SF - South Africa	10	WD AF
68	EE - Eastern Europe	11	WD AS
67	SV - Soviet Union	12	WD EU AS
152	CH - P.R. China	13	WD AS
8	MX - Mexico	14	WD WH AS
190	CA - C. Amer. & Caribb.	15	WD AS
31	AR - Argentina	16	WD WH
32	BZ - Brazil	17	WD WH
43	VE - Venezuela	18	WD WH
199	LA - O. Latin America	19	WD
116	NG - Nigeria	20	WD AF
200	AF - O. Sub.S. Africa	21	WD
97	EG - Egypt	22	WD AF
191	MP - M.E. & N.A. Oil P.	23	WD
192	MO - M.E. & N.A. Other	24	WD
176	ND - India	25	WD AS
201	OS - O. South Asia	26	WD
164	DO - Indonesia	27	WD AS
170	TH - Thailand	28	WD AS
167	ML - MaLaysia	29	WD AS
168	PH - Philippines	30	WD AS
202	SA - O. S.E. Asia	31	WD
159	SK - South Korea	32	WD AS
160	TW - Taiwan	33	WD AS
203	EA - O. East Asia	34	WD
209	RW - Rest of World - WD	35	WD
193	DA - Dev. Asian Imp.	36	EU (from SW89, an aggregate of WD)
194	DE - Dev. Exporters	37	EU (from SW89, an aggregate of WD)
45	BL - Belgium-Lux.	38	EU
46	DN - DeNmark	39	EU
47	FR - FRance	40	EU
48	GC - GreeCe	41	EU
49	IR - IReland	42	EU
50	IT - ITaly	43	EU
51	NT - NeTherlands	44	EU
52	PT - PorTugal	45	EU
53	SP - SPain	46	EU
54	UK - United Kingdom	47	EU
56	WG - West Germany	48	EU
57	GD - German Dem. Rep.	49	EU
59	AT - AusTria	50	EU
61	FN - FiNland	51	EU
64	NO - NOrway	52	EU
65	SW - SWeden	53	EU
66	SZ - SwitZerland	54	EU
195	OW - O. West. Europe	55	EU
69	AB - AlBania	56	EU
70	BI - BulgarIa	57	EU
71	CZ - CZeckoslovakia	58	EU
72	HU - HUngary	59	EU

Note that many of these country TSView data sets and models are used in several SWOPSIM global 1989 models.

Number in FAS 189 Country PS&D Database	DWOPSIM/SWOPSIM TSView Database 2 Letter Code and Country/Region Name	TSView ORDER	SWOPSIM 2 Digit Global Model Code for Model Containing Country/Region WD = WorlD, EU = EUrope, WH = Western Hemisphere, AS = ASia and Pacific Rim, AF = AFrica
73	PL - PoLand	60	EU
74	RM - RoMania	61	EU
75	YU - YUgoslavia	62	EU
92	TK - TurKey	63	EU
95	NF - North AFrica	64	EU
204	OM - O. Middle East	65	EU
210	R1 - ROW for EU model	66	EU
10	BE - BElize	67	WH
11	CR - Costa Rica	68	WH
12	ES - El Salvador	69	WH
13	GT - GuaTemala	70	WH
14	HO - HOnduras	71	WH
15	NI - NIcaragua	72	WH
16	PA - PAnama	73	WH
18	BH - BaHamas	74	WH
19	BA - BArbados	75	WH
20	BD - BermuDa	76	WH
21	CU - CUba	77	WH
22	DR - Dominican Republic	78	WH
23	GU - GUadaloupe	79	WH
24	HA - HAiti	80	WH
25	JM - JaMaica	81	WH
26	MA - MArtinique	82	WH
27	NN - Netherlands ANtil.	83	WH
28	SC - St. LuCia	84	WH
29	ST - ST. Vincent	85	WH
30	TT - Trinidad-Tobago	86	WH
34	BO - BOlivia	87	WH
35	CL - ChiLe	88	WH AS
36	CO - COlumbia	89	WH AS
37	ED - EcuaDor	90	WH AS
38	GY - GuYana	91	WH
39	PR - PaRaguay	92	WH
40	PE - PEru	93	WH AS
41	SU - SUrinam	94	WH
42	UR - URuguay	95	WH
211	R2 - ROW for WH model	96	WH
153	MN - MoNgolia	97	AS
158	NK - North Korea	98	AS
162	BR - BRunai	99	AS
163	BU - BUrma	100	AS
165	KR - Kymer Republic	101	AS
166	LO - LaOs	102	AS
169	SN - SiNgapore	103	AS
171	VT - VieTnam	104	AS
173	AH - AfgHanistan	105	AS
174	BG - BanGladesh	106	AS
178	PK - PaKistan	107	AS
179	SL - Sri Lanka	108	AS
196	OA - Other Asia	109	AS
212	R3 - ROW for ASia model	110	AS
197	OE - Other Europe	111	AF
198	AS - ASia	112	AF
96	AL - ALgeria	113	AF
98	LY - LibYa	114	AF
99	MC - MoroCco	115	AF
100	TN - TuNisia	116	AF
104	CM - CaMeroon	117	AF
108	GH - GHana	118	AF
109	GN - GuiNea	119	AF
111	IC - Ivory Coast	120	AF
112	LB - LiBeria	121	AF
117	SG - SeneGal	122	AF
205	AW - O. West Africa	123	AF
126	ZR - ZaiRe	124	AF
206	CF - Central AFrica	125	AF

Number FAS 189 Country PS&D Database	DWOPSIM/SWOPSIM TSView Database 2 Letter Code and Country/Region Name	TSView ORDER	SWOPSIM 2 Digit Global Model Code for Model Containing Country/Region WD = WorlD, EU = EUrope, WH = Western Hemisphere, AS = ASia and Pacific Rim, AF = AFrica
130	EP - EthioPia	126	AF
131	KY - KenYa	127	AF
133	SM - SoMalia	128	AF
134	SD - SuDan	129	AF
135	TZ - TanZania	130	AF
136	UG - UGanda	131	AF
207	AE - O. East Africa	132	AF
138	AG - AnGola	133	AF
139	BT - BoTswana	134	AF
141	LH - LesotHo	135	AF
142	MG - MadaGascar	136	AF
143	MW - MalaWi	137	AF
145	MZ - MoZambique	138	AF
148	WZ - SWaZiland	139	AF
149	ZA - ZAmbia	140	AF
150	ZB - ZimBabwe	141	AF
208	OF - O. Southern AFrica	142	AF
213	R4 - ROW for Africa	143	AF

DWOPSIM/SWOPSIM Aggregate Region Code	FAS PS&D Country Code Numbers Included in the Aggregate Regions
CA	9 17
MP	77 79 80 83 88 89 90 91 93 96 98 100
MO	78 81 82 84 85 92 99
CP	67 68 152
DA	155 157 158 159 160
DE	31 32 164 167 168 170
OW	60 62 63
OA	155 157 175 177 182 183 184 185 186 187 188 189
OE	58 67 68
AS	152 153 154 161 172
LA	34 35 36 37 38 39 40 41 42
AF	102 103 104 105 106 107 108 109 110 111 112 113 114 115 117 118 119 120 127 138 139 140 141 142 143 144 145 146 147 148 149 150
OS	173 174 175 177 178 179 182 183 184 185 186 187 188 189
SA	162 163 165 166 169 171
EA	155 157 158
OM	77 78 79 80 81 82 83 84 85 88 89 90 91 93
AW	102 103 105 106 107 110 113 114 115 118 119
CF	121 122 123 124 125
AE	128 129 132
OF	140 144 146 147

DWOPSIM /SWOPSIM ROW Code	FAS PS&D Country Code Numbers Subtracted from the World Total (the first code which is 1)
RW	6 7 8 31 32 43 44 58 67 68 97 116 151 152 156 159 160 164 167 168 170 176 180 181 190 191 192 199 200 201 202 203
R1	6 7 45 46 47 48 49 50 51 52 53 54 56 57 59 61 64 65 66 67 69 70 71 72 73 74 75 92 95 156 180 181 193 194 195 204
R2	6 7 8 10 11 12 13 14 15 16 18 19 20 21 22 23 24 25 26 28 29 30 31 32 34 35 36 37 38 39 40 41 42 43 44 156
R3	6 7 8 35 36 37 40 44 58 67 68 152 153 156 158 159 160 162 163 164 165 166 167 168 169 170 171 173 174 176 178 179 180 181 190 196
R4	6 44 96 97 98 99 100 104 108 109 111 112 116 117 126 130 131 133 134 135 136 138 139 141 142 143 145 148 149 150 151 197 198 205 206 207 208

The program TSSWOP calculates values for the ROW (Rest-Of-World) region of each world model by subtracting values for each FAS country from the FAS world value (FAS code = 1). The subtraction is done with a TSVIEW utility controlled by a DOS batch program. If the TS data files are updated, TSSWOP can be run again to recalculate the SWOPSIM intermediate database values.

FAS Database Country/Region	FAS PS&D Country Number Code
WORLD	1
FOREIGN	2
CENTRALLY PLANNED	3
NICs	4
LESS DEVELOPED	5
UNITED STATES	6
CANADA	7
MEXICO	8
CENTRAL AMERICA	9
BELIZE	10
COSTA RICA	11
EL SALVADOR	12
GUATEMALA	13
HONDURAS	14
NICARAGUA	15
PANAMA	16
CARIBBEAN	17
BAHAMAS	18
BARBADOS	19
BERMUDA	20
CUBA	21
DOMINICAN REPUBLIC	22
GUADELOUPE	23
HAITI	24
JAMAICA	25
MARTINIQUE	26
NETHERLAND ANTILLES	27
ST LUCIA	28
ST VINCENT	29
TRINIDAD-TOBAGO	30
ARGENTINA	31
BRAZIL	32
OTHER SOUTH AMERICA	33
BOLIVIA	34
CHILE	35
COLOMBIA	36
ECUADOR	37
GUYANA	38
PARAGUAY	39
PERU	40
SURINAM	41
URUGUAY	42
VENEZUELA	43
EC-12	44
BELGIUM-LUX	45
DENMARK	46
FRANCE	47
GREECE	48
IRELAND	49
ITALY	50
NETHERLANDS	51
PORTUGAL	52
SPAIN	53
UK	54
UNITED GERMANY	55
WEST GERMANY	56
EAST GERMANY	57
OTHER WESTERN EUROPE	58
AUSTRIA	59
FAEROE ISLANDS	60
FINLAND	61
ICELAND	62
MALTA	63
NORWAY	64
SWEDEN	65
SWITZERLAND	66
SOVIET UNION	67
EASTERN EUROPE	68
ALBANIA	69
BULGARIA	70
CZECHOSLOVAKIA	71
HUNGARY	72

These are the countries and regions in the 189 country/ region FAS PS&D database kept in the TSView data system. The numerical codes are to the right of the country/ region names.

FAS Database Country/Region	FAS PS&D Country Number Code
POLAND	73
ROMANIA	74
YUGOSLAVIA	75
MIDDLE EAST	76
BAHRAIN	77
CYPRUS	78
IRAN	79
IRAQ	80
ISRAEL	81
JORDAN	82
KUWAIT	83
LEBANON	84
UNITED YEMEN	85
NORTH YEMEN-SANA	86
SOUTH YEMEN-ADEN	87
OMAN	88
QATAR	89
SAUDI ARABIA	80
SYRIA	91
TURKEY	92
UNITED ARAB EMIRATES	93
SUBSAHARAN AFRICA	94
NORTH AFRICA	95
ALGERIA	96
EGYPT	97
LIBYA	98
MOROCCO	99
TUNISIA	100
WEST AFRICA	101
BENIN	102
BURKINA	103
CAMEROON	104
CAPE VERDE	105
CHAD	106
GAMBIA	107
GHANA	108
GUINEA	109
GUINEA-BISSAU	110
IVORY COAST	111
LIBERIA	112
MALI	113
MAURITANIA	114
NIGER	115
NIGERIA	116
SENEGAL	117
SIERRE LEONE	118
TOGO	119
CENTRAL AFRICA	120
CENTRAL AFRICAN REPUBLIC	121
CONGO	122
EQUATORIAL GUINEA	123
GABON	124
SAO TOME & PRINCIPE	125
ZAIRE	126
EAST AFRICA	127
BURUNDI	128
DJIBOUTI	129
ETHIOPIA	130
KENYA	131
RWANDA	132
SOMALIA	133
SUDAN	134
TANZANIA	135
UGANDA	136
SOUTHERN AFRICA	137
ANGOLA	138
BOTSWANA	139
COMORO ISLANDS	140
LESOTHO	141
MADAGASCAR	142
MALAWI	143
MAURITIUS	144

FAS Database Country/Region	FAS PS&D Country Number Code
MOZAMBIQUE	145
REUNION	146
SEYCHELLES	147
SWAZILAND	148
ZAMBIA	149
ZIMBABWE	150
REPUBLIC OF SOUTH AFRICA	151
CHINA	152
OUTER MONGOLIA	153
EAST ASIA	154
HONG KONG	155
JAPAN	156
MACAO	157
NORTH KOREA	158
SOUTH KOREA	159
TAIWAN	160
SOUTHEAST ASIA	161
BRUNEI	162
BURMA	163
INDONESIA	164
KHMER REPUBLIC	165
LAOS	166
MALAYSIA	167
PHILIPPINES	168
SINGAPORE	169
THAILAND	170
VIETNAM	171
SOUTH ASIA	172
AFGHANISTAN	173
BANGLADESH	174
BHUTAN	175
INDIA	176
NEPAL	177
PAKISTAN	178
SRI LANKA	179
AUSTRALIA	180
NEW ZEALAND	181
PAPUA NEW GUINEA	182
FIGI	183
WEST SAMOA	184
NEW CALEDONIA	185
TONGA	186
BRITISH SOLOMON ISLANDS	187
GILBERT & ELLICE ISLANDS	188
NEW HEBRIDES	189

Commodity in FAS 189 Country PS&D Database	Product/Product Group Contained in TSView DWOPSIM/SWOPSIM Database		* DWOPSIM/SWOPSIM Product Group Calculated from FAS PS&D Commodities:	
BEEFVEAL	BF - BeeF and veal	1		
PORK	PK - PorK	2		
LAMBMUTT	ML - Mutton and Lamb	3		
TOTPLTRY	PM - Poultry Meat	4		
EGGS	PE - Poultry Eggs	5		
FLUIDMLK	DM - Dairy - fresh Milk	6		
BUTTER	DB - Dairy - Butter	7		
CHEESE	DC - Dairy - Cheese	8		
NFDMILK	DP - Dairy - Powder	9		
WHEAT	WH - WHeat	10		
CORN	CN - CorN	11		-- = Standard 22 Product Model
* Calc.	CG - other Coarse Grains	12	COARSEGR-CORN	
RICE	RI - RIce	13		
SOYBEANS	SB - SoyBeans	14		
SOYMEAL	SM - SoyMeal	15		
SOYOIL	SO - SoyOil	16		
* Calc.	OS - Other oilSeeds	17	TOTSEEDS-SOYBEANS	
* Calc.	OM - Other Meals	18	TOTMEALS-SOYMEAL	
* Calc.	OO - Other Oils	19	TOTOILS-SOYOIL	
COTTON	CT - CoTton	20		
SUGAR	SU - SUgar	21		
TOBUNMFG	TB - ToBacco	22		
DRYBEANS	BN - dry BeaNs	23		- = Other Products in TSView SWOPSIM Database
COFFEE	CF - green CoFfee	24		
HIDESKIN	HS - Hides and Skins	25		
CATTLEIN	CI - Cattle INventory	26		
SWINEINV	PI - Pig Inventory	28		
SHEEPINV	SI - Sheep Inventory	28		
TALLOWGR	TG - Tallow and Grease	29		
SORGHUM	SG - SorGhum	30		--Subset of standard 22 product Model
BARLEY	BY - BarleY	31		
OATS	OT - OaTs	32		
* Calc	OC - Other Coarse grains	33	COARSEGR-CORN-SORGHUM-OATS	
RAPESEED	RS - RapeSeed	34		
SUNSEED	US - sUnflower Seed	35		
PEANUTS	NS - peaNutS	36		
* Calc	TS - oTher oilSeeds	37	TOTSEEDS-SOYBEANS-RAPESEED-SUNSEED-PEANUTS	

Commodity in FAS 189 Country PS&D Database	Product/Product Group Contained in TSView DWOPSIM/SWOPSIM Database		* DWOPSIM/SWOPSIM Product Group Calculated from FAS PS&D Commodities:	
RAPEMEAL	RM - Rapeseed Meal	38		
SUNMEAL	UM - sUnflower seed Meal	39		
PNUTMEAL	NM - peaNut Meal	40		
* Calc	TM - oTher Meals	41	TOTMEALS-SOYMEAL-RAPEMEAL-SUNMEAL-PNUTMEAL	--Subset of standard 22 product Model
RAPEOIL	RO - Rapeseed Oil	42		
SUNOIL	UO - sUnflower seed Oil	43		
PNUTOIL	NO - peaNut Oil	44		
* Calc	TO - oTher Oils	45	TOTOILS-SOYOIL-RAPEOIL-SUNOIL-PNUTOIL	
TURKEY	TK - Turkey meat	46		
* Calc	BM - Broilers and other meat	47	TOTPLTRY-TURKEY	

Appendix 2--Documentation of the DWOPSIM Demonstration Model TIME

This appendix gives sample documentation for one model spreadsheet in TIME (USBF). The user can use DWOPSIM programs to print other spreadsheets (see fig. 1, page 2, which shows the spreadsheet structure of TIME). Documentation of the data in a model spreadsheet is printed out (or printed to disk) by the program DWOPBOUT. If a model spreadsheet contains equations, the program DWOPEOUT will print documentation of the equations.[1]

Data documentation: The data in TIME are history for 1960-91 and projected to 2020. Although the model spreadsheets actually contain projected data up to 2020, DWOPBOUT prints data from 1960 to 2005. The structure of the model spreadsheets is part of the model design, and it is a matter of judgment of how large a spreadsheet is needed. The user can obtain full printouts of the model spreadsheets by using normal spreadsheet printing techniques.

Equation documentation: Documentation of the equations coded in the equation spreadsheet is the standard output of the DWOPEOUT program. DWOPEOUT produces three pages for each model equation spreadsheet. The first two pages are a matrix of cells in the spreadsheet. The rows contain information about terms in an equation; the name of the variable explained is on the left along with the letter of its home column in the model spreadsheet. Some rows have no names and no information in cells--these are just extra rows that might contain equation information in more complex models. The columns in the model equation spreadsheet represent the explanatory variables in the equations. If there is no variable name under a column, there is no explanatory variable in that column in the spreadsheet. If cells under a variable name are empty, that means that the column variable is not used in an equation represented by the cell row. When a cell has contents, notation in the cell gives the mathematical representation of how the column variable (represented by the "v") is included in the equation represented by the row. Parameters are also included in cells. For example, in many of the TIME model cells, one will see elasticities or other parameters as part of the cell notation. The bottom half of the second page (right side) gives some notes explaining the nomenclature used for variables in the model TIME.

The third page of the equation documentation gives a "readable" version of the equations themselves.[2] The top part of the page gives the variable codes and their descriptions, units, and home spreadsheet column. The bottom half of the page gives the equations themselves written using DWOPSIM variable nomenclature from the model TIME. The spacing of the variables and equations follows their row position in the equation spreadsheet. Variables explained by equations are marked by an "X". Other variables will be exogenous variables, formula driven identities, or variables from other spreadsheets.

---

[1]Note that in printout of data from the model spreadsheets, cells of columns containing no data are marked by dots "." A more complex model could add variables in these columns. "N/A"s mark cells where data are "not available" for particular years. In the model TIME, this is the case with support information which exists for a shorter period than quantity data.

[2]This is useful because the equations in spreadsheet cells use only cell notation to designate variables. This is efficient for computation but makes it difficult to read the equations. This third page of the DWOPEOUT printout provides a "readable" version of equations for the convenience of the user.

Col.>	B	C	D	E	F	G	H	I	J	K	L	M	N	O	Row
USBF	SLa	Slaug	PRoducti	IMports	EXports	CoNsumpt	End.Stoc								1
Unit	number	KG/an	1000 MT	1000 MT	1000 MT	1000 MT	1000 MT								2
SSWC	BFnSL	BFrSW	BFqPR	BFqIM	BFqEX	BFqCN	BFqES								3
Year															TIME
1960	34644	.2076839	7195	352	4	7543	83	.	.	.	.	.	.	.	5
	34551	.2149287	7426	470	15	7868	96	.	.	.	.	.	.	.	6
	34768	.2069432	7195	352	16	7535	92	.	.	.	.	.	.	.	7
	35274	.2105233	7426	470	16	7842	130	.	.	.	.	.	.	.	8
	39310	.1885271	7411	653	15	8030	149	.	.	.	.	.	.	.	9
1965	40959	.2186821	8957	427	24	9387	122	.	.	.	.	.	.	.	10
	41036	.2280924	9360	546	18	9866	144	.	.	.	.	.	.	.	11
	40407	.2358750	9531	602	19	10128	130	.	.	.	.	.	.	.	12
	41034	.2389238	9804	689	17	10469	137	.	.	.	.	.	.	.	13
	40584	.2439878	9902	744	15	10603	165	.	.	.	.	.	.	.	14
1970	39559	.2553907	10103	824	18	10917	157	.	.	.	.	.	.	.	15
	39730	.2563302	10184	797	24	10944	170	.	.	.	.	.	.	.	16
	39335	.2637346	10374	905	28	11249	172	.	.	.	.	.	.	.	17
	36506	.2686682	9808	917	41	10647	209	.	.	.	.	.	.	.	18
	40528	.2643851	10715	747	29	11453	189	.	.	.	.	.	.	.	19
1975	46870	.2404737	11271	808	24	12080	164	.	.	.	.	.	.	.	20
	48726	.2496819	12166	950	41	13024	215	.	.	.	.	.	.	.	21
	48073	.2463753	11844	890	47	12754	148	.	.	.	.	.	.	.	22
	44272	.2547660	11279	1053	74	12160	246	.	.	.	.	.	.	.	23
	36932	.2687101	9924	1103	78	10982	213	.	.	.	.	.	.	.	24
1980	36795	.2717217	9998	946	80	10877	200	.	.	.	.	.	.	.	25
	38151	.2713952	10354	799	100	11097	156	.	.	.	.	.	.	.	26
	39264	.2655104	10425	889	115	11176	179	.	.	.	.	.	.	.	27
	40136	.2677895	10748	873	125	11476	199	.	.	.	.	.	.	.	28
	41259	.2648877	10929	838	152	11594	220	.	.	.	.	.	.	.	29
1985	40048	.2745955	10997	948	151	11819	195	.	.	.	.	.	.	.	30
	41046	.2751060	11292	978	239	12036	190	.	.	.	.	.	.	.	31
	38792	.2805733	10884	1040	277	11660	177	.	.	.	.	.	.	.	32
	37889	.2871546	10880	1091	313	11641	194	.	.	.	.	.	.	.	33
	36329	.2926863	10633	988	464	11197	154	.	.	.	.	.	.	.	34
1990	35245	.2969215	10465	1068	456	11048	183	.	.	.	.	.	.	.	35
	35650.00	.2969144	10585.00	1030.000	472.0000	11181.00	145.0000	.	.	.	.	.	.	.	36
.	35831.48	.2969144	10638.88	1046.383	474.4028	11358.84	145.7381	.	.	.	.	.	.	.	37
.	36106.92	.2969144	10720.67	1062.706	478.0496	11536.04	146.8585	.	.	.	.	.	.	.	38
	36309.12	.2969144	10780.70	1078.970	480.7267	11712.59	147.6809	.	.	.	.	.	.	.	39
1995	36512.41	.2969144	10841.06	1095.173	483.4182	11888.48	148.5077	.	.	.	.	.	.	.	40
	36716.84	.2969144	10901.76	1110.946	486.1247	12059.70	149.3392	.	.	.	.	.	.	.	41
.	36920.13	.2969144	10962.12	1126.921	488.8163	12233.11	150.1660	.	.	.	.	.	.	.	42
.	37124.05	.2969144	11022.67	1143.081	491.5161	12408.54	150.9954	.	.	.	.	.	.	.	43
	37326.49	.2969144	11082.78	1159.429	494.1965	12586.00	151.8188	.	.	.	.	.	.	.	44
2000	37529.97	.2969144	11143.19	1175.966	496.8905	12765.52	152.6464	.	.	.	.	.	.	.	45
	37732.10	.2969144	11203.21	1197.313	499.5667	12997.24	153.4686	.	.	.	.	.	.	.	46
.	37935.22	.2969144	11263.51	1219.039	502.2559	13233.08	154.2947	.	.	.	.	.	.	.	47
.	38139.35	.2969144	11324.12	1241.150	504.9586	13473.10	155.1250	.	.	.	.	.	.	.	48
	38341.93	.2969144	11384.27	1263.652	507.6407	13717.37	155.9489	.	.	.	.	.	.	.	49
2005	38545.47	.2969144	11444.71	1286.551	510.3355	13965.95	156.7768	.	.	.	.	.	.	.	50

Col.>	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	Row
USBF				BF-PI.pr	BF-CI.pr	Net Trad	P.S.wedg	C.S.wedg	E.S.wedg	I.S.wedg	PRod.shf	CoNs.shf			1
Unit				89.US$/M	89.US$/M	1000 MT	89.US$/M	89.US$/M	89.US$/M	89.US$/M	Percent	Number			2
SSWC				BFpPI	BFpCI	BFqNT	BFwPS	BFwCS	BFwES	BFwIS	BFfPR	BFfCN			3
Year															TIME
1960	.	.	.	N/A	N/A	-348	N/A	N/A	N/A	N/A	N/A	N/A	.	.	5
	.	.	.	N/A	N/A	-455	N/A	N/A	N/A	N/A	N/A	N/A	.	.	6
	.	.	.	N/A	N/A	-336	N/A	N/A	N/A	N/A	N/A	N/A	.	.	7
	.	.	.	N/A	N/A	-454	N/A	N/A	N/A	N/A	N/A	N/A	.	.	8
	.	.	.	N/A	N/A	-638	N/A	N/A	N/A	N/A	N/A	N/A	.	.	9
1965	.	.	.	N/A	N/A	-403	N/A	N/A	N/A	N/A	N/A	N/A	.	.	10
	.	.	.	N/A	N/A	-528	N/A	N/A	N/A	N/A	N/A	N/A	.	.	11
	.	.	.	N/A	N/A	-583	N/A	N/A	N/A	N/A	N/A	N/A	.	.	12
	.	.	.	N/A	N/A	-672	N/A	N/A	N/A	N/A	N/A	N/A	.	.	13
	.	.	.	N/A	N/A	-729	N/A	N/A	N/A	N/A	N/A	N/A	.	.	14
1970	.	.	.	N/A	N/A	-806	N/A	N/A	N/A	N/A	N/A	N/A	.	.	15
	.	.	.	N/A	N/A	-773	N/A	N/A	N/A	N/A	N/A	N/A	.	.	16
	.	.	.	N/A	N/A	-877	N/A	N/A	N/A	N/A	N/A	N/A	.	.	17
	.	.	.	N/A	N/A	-876	N/A	N/A	N/A	N/A	N/A	N/A	.	.	18
	.	.	.	N/A	N/A	-718	N/A	N/A	N/A	N/A	N/A	N/A	.	.	19
1975	.	.	.	N/A	N/A	-784	N/A	N/A	N/A	N/A	N/A	N/A	.	.	20
	.	.	.	N/A	N/A	-909	N/A	N/A	N/A	N/A	N/A	N/A	.	.	21
	.	.	.	N/A	N/A	-843	N/A	N/A	N/A	N/A	N/A	N/A	.	.	22
	.	.	.	N/A	N/A	-979	N/A	N/A	N/A	N/A	N/A	N/A	.	.	23
	.	.	.	N/A	N/A	-1025	N/A	N/A	N/A	N/A	N/A	N/A	.	.	24
1980	.	.	.	N/A	N/A	-866	N/A	N/A	N/A	N/A	N/A	N/A	.	.	25
	.	.	.	N/A	N/A	-699	N/A	N/A	N/A	N/A	N/A	N/A	.	.	26
	.	.	.	N/A	N/A	-774	N/A	N/A	N/A	N/A	N/A	N/A	.	.	27
	.	.	.	N/A	N/A	-748	N/A	N/A	N/A	N/A	N/A	N/A	.	.	28
	.	.	.	N/A	N/A	-686	N/A	N/A	N/A	N/A	N/A	N/A	.	.	29
1985	.	.	.	N/A	N/A	-797	N/A	N/A	N/A	N/A	N/A	N/A	.	.	30
	.	.	.	N/A	N/A	-739	N/A	N/A	N/A	N/A	N/A	N/A	.	.	31
	.	.	.	N/A	N/A	-763	N/A	N/A	N/A	N/A	N/A	N/A	.	.	32
	.	.	.	N/A	N/A	-778	N/A	N/A	N/A	N/A	N/A	N/A	.	.	33
	.	.	.	2636	4779	-524	N/A	14	N/A	-44	N/A	N/A	.	.	34
1990	.	.	.	2559.223	4639.806	-612	N/A	13.59223	N/A	-42.7184	N/A	N/A	.	.	35
	.	.	.	2555.249	4575.232	-558.000	0	13.19634	0	-41.4742	0	0	.	.	36
.	.	.	.	2525.541	4545.908	-720.694	0	12.81198	0	-40.2662	0	0	.	.	37
.	.	.	.	2496.068	4516.808	-816.491	0	12.43882	0	-39.0934	0	0	.	.	38
	.	.	.	2466.829	4487.932	-932.705	0	12.07652	0	-37.9548	0	0	.	.	39
1995	.	.	.	2437.824	4459.278	-1048.24	0	11.72478	0	-36.8493	0	0	.	.	40
	.	.	.	2409.050	4430.846	-1158.77	0	11.38328	0	-35.7760	0	0	.	.	41
.	.	.	.	2380.408	4402.536	-1271.82	0	11.05173	0	-34.7340	0	0	.	.	42
.	.	.	.	2351.997	4374.446	-1386.70	0	10.72983	0	-33.7223	0	0	.	.	43
	.	.	.	2323.814	4346.576	-1504.05	0	10.41731	0	-32.7401	0	0	.	.	44
2000	.	.	.	2295.861	4318.926	-1623.15	0	10.11390	0	-31.7865	0	0	.	.	45
	.	.	.	2268.135	4291.495	-1794.86	0	9.819318	0	-30.8607	0	0	.	.	46
.	.	.	.	2240.636	4264.282	-1970.40	0	9.533319	0	-29.9619	0	0	.	.	47
.	.	.	.	2213.363	4237.287	-2149.81	0	9.255649	0	-29.0892	0	0	.	.	48
	.	.	.	2186.316	4210.510	-2333.92	0	8.986067	0	-28.2419	0	0	.	.	49
2005	.	.	.	2159.494	4183.949	-2522.07	0	8.724337	0	-27.4193	0	0	.	.	50

USBFEQ	SLaughter	Slaught.Wt.	PRoduction	IMports	EXports	CoNsumption	End.Stocks
Base row> 36	number	KG/animal	1000 MT	1000 MT	1000 MT	1000 MT	1000 MT
Variable-(SSWC)->	USBFrSL	USBFrSW	USBFqPR	USBFqIM	USBFqEX	USBFqCN	USBFqES
Variable column-->	B	C	D	E	F	G	H
BFrSL B			*v				
BFrSW C	(1/v)		*v				
BFqPR D							
BFqIM E						*v	
BFqEX F			*v				
BFqCN G							
BFqES H			*v				
BFpPI S							
BFpPI1 S							
BFpPI2 S							
BFpCI T							
BFqNT U							
BFwPS V							
BFwCS W							
BFwES X							
BFwIS Y							
BFePF Z							
BFfCF AA							

USBFEQ	BF-PI.price	BF-PI.price	BF-PI.price	BF-CI.price	Net Trade
Base row> 36	89.US$/MT	89.US$/MT	89.US$/MT	89.US$/MT	1000 MT
Variable-(SSWC)->	USBFpPI	USBFpPI1	USBFpPI2	USBFpCI	USBFqNT
Variable column-->	S	S	S	T	U
BFrSL B					
BFrSW C					
BFqPR D	*v^.1	*v^.3	*v^.2		
BFqIM E					
BFqEX F					
BFqCN G				*v^-.7	
BFqES H					
BFpPI S					
BFpPI1 S					
BFpPI2 S					
BFpCI T	v				
BFqNT U					
BFwPS V					
BFwCS W					
BFwES X					
BFwIS Y					
BFePF Z					
BFfCF AA					

USBFEQ	P.S.wedge	C.S.wedge	E.S.wedge	I.S.wedge	Prod.shFt.	Cons.shFt.				
Base row> 36	89.US$/MT	89.US$/MT	89.US$/MT	89.US$/MT	Number	Number				
Variable-(SSVVC)->	USBFwPS	USBFwCS	USBFwES	USBFwIS	USBFePF	USBFfCF				
Variable column-->	V	W	X	Y	Z	AA				
BFrSL B										
BFrSW C										
BFqPR D					*(1+v)					
BFqIM E										
BFqEX F										
BFqCN G						*(1+v)				
BFqES H										
BFpPI S				-v						
BFpPI1 S										
BFpPI2 S										
BFpCI T		-v								
BFqNT U										
BFwPS V										
BFwCS W										
BFwES X										
BFwIS Y										
BFePF Z										
BFfCF AA										

USBFEQ	CN-CN.price	WD Ref.P.	TiMe trend	real INcome	PoPulation
Base row> 36	89.US$/MT	89.US$/MT	count	M.89.US$	1000s
Variable-(SSVVC)->	USCNpCN	WDpRP	WDeTM	USeIN	USePP
Variable column-->	T	C	B	B	C
BFrSL B					
BFrSW C					
BFqPR D	*v^-.05		*(1+.0116)^v		
BFqIM E					
BFqEX F					
BFqCN G				*v^.16	*v^(1-.16)
BFqES H					
BFpPI S		+v			
BFpPI1 S					
BFpPI2 S					
BFpCI T					
BFqNT U					
BFwPS V					
BFwCS W					
BFwES X					
BFwIS Y					
BFePF Z					
BFfCF AA					

Model = TIME

--------Notes--------

Variable names have form: SSVVC where:

SS = SpreadSheet for country & product, country/region, or world (prices):

v = Variable type:

a - Area
n - Number
q - Quantity,
r - Ratio
p - Price,
w - price Wedge
f - shiFt percent
l - vaLue

VC# = Variable name Code and # = var. lag:

HV - area HarVested
PR - PRoduction
IM - IMports
EX - EXports
NT - Net Trade
CN - CoNsumption
ES - Ending Stocks
FD - FeeD consump.
OC - Other Consump.
PI - Producer Incentive price
CI - Consumer Incentive price
RF - Reference Price
IN - INcome
PP - PoPulation
TM - TiMe

Variables and Equations in spreadsheet--> USBF Model = TIME

Variable	Eq.(x)	Variable description (unit), spreadsheet column
USBFnSL	x	SLaughter (number), col. B
USBFrSW	x	Slaught.Wt. (KG/animal), col. C
USBFqPR	x	PRoduction (1000 MT), col. D
USBFqIM	x	IMports (1000 MT), col. E
USBFqEX	x	EXports (1000 MT), col. F
USBFqCN	x	CoNsumption (1000 MT), col. G
USBFqES	x	End.Stocks (1000 MT), col. H
USBFpPI	x	BF-PI.price (89.US$/MT), col. S
USBFpPI1		BF-PI.price (89.US$/MT), col. S
USBFpPI2		BF-PI.price (89.US$/MT), col. S
USBFpCI	x	BF-CI.price (89.US$/MT), col. T
USBFqNT		Net Trade (1000 MT), col. U
USBFwPS		P.S.wedge (89.US$/MT), col. V
USBFwCS		C.S.wedge (89.US$/MT), col. W
USBFwES		E.S.wedge (89.US$/MT), col. X
USBFwIS		I.S.wedge (89.US$/MT), col. Y
USBFePF		Prod.shFt. (Number), col. Z
USBFfCF		Cons.shFt. (Number), col. AA
USCNpCN		CN-CN.price (89.US$/MT), col. T
WDpRP		WD Ref.P. (89.US$/MT), col. C
WDeTM		TiMe trend (count), col. B
USeIN		real INcome (M.89.US$), col. B
USePP		PoPulation (1000s), col. C

Var. Equation.....

```
USBFnSL = Constant*USBFqPR
USBFrSW = Constant+(1/USBFnSL)*USBFqPR
USBFqPR = Constant*USBFpPI^.1*USBFpPI1^.3*USBFpPI2^.2*(1+USBFePF)*USCNpCN^-.05*(1+.0116)^WDeTM
USBFqIM = Constant*USBFqCN
USBFqEX = Constant*USBFqPR
USBFqCN = Constant*USBFpCI^-.7*(1+USBFfCF)*USeIN^.16*USePP^(1-.16)
USBFqES = Constant*USBFqPR

USBFpPI = Constant+-USBFwIS+WDpRP

USBFpCI = Constant+USBFpPI-USBFwCS
```

Appendix 3--Annotated Listings of DWOPSIM Computer Programs

DWOPSIM computer programs and routines are relatively simple. When tasks require spreadsheet formats or data manipulations, spreadsheet templates are created. The C:\DWOPSIM subdirectory contains several spreadsheet templates, which are called and used by DOS batch programs, usually with the same name as the spreadsheet. Some spreadsheets are complex such as the one that actually writes DWOPSIM equations. But all programming, whether in DOS batch programs or in spreadsheets, is open to the user.

DOS batch programs control DWOPSIM operations and write spreadsheet macros. The listings that follow are the key DOS batch programs that make up DWOPSIM. In some cases, annotation has been added to the documentation already contained in the programs. The name of each batch program is highlighted in **bold** print. Almost all DWOPSIM DOS batch programs have explanatory screens. If the user does not know what a batch program does or what parameters it requires, typing the name of the program will bring up the explanatory screen. In the following pages, the material in the explanatory screens will be seen at the beginning of the batch programs, produced by ECHO commands on the left.

Most DWOPSIM programs apply to all models; the first four characters of their names are DWOP. However, three important DOS batch programs must be created for each model. For the model TIME, they have the names DWOPTIME, TIMESOLVE, and TIMEAGAN. These programs can be created for a new model by selective text editing of the programs associated with the model TIME. Certain parts of the batch programs and/or associated spreadsheets must conform with product and country codes in a new model. And lines must be added if the new model has more countries or commodities than the model TIME. The creation of these three "customized" batch program defines a new model much in the way that a new master model file defined a new model in SWOPSIM.

The <u>first program</u> to be created (DWOPTIME for the model TIME) tells DWOPSIM what country data to get from the TSView database for the world model. It also defines the rest-of-world region as the world minus the countries/regions explicitly defined in the model. When this program is run, the user gets a globally balanced set of commodity data for the commodity selected. Balanced means that rest-of-world data from the TSView database are adjusted so that world supply equals world demand, world net trade equals zero, and world imports equal world exports for every year in the historical data. The "balancing" rule is contained in a spreadsheet template DWOPBLRW.CAL. This spreadsheet has to be consistent with the data modeled. For example, in the model TIME, data include stock changes and, therefore, world balances must account for stock changes. The <u>second program</u> to be created (TIMESOLV) tells DWOPSIM what commodity spreadsheets need to be collected into memory to form a world model. The program is called by the generalized DWOPSOLV program. The <u>third program</u> is TIMEAGAN, which allows repeated scenario analysis once DWOPSOLV has been run.

All of the programs have error traps that stop the program and tell the user if some important spreadsheet or program or piece of entry information is missing. When a program successfully completes its task, the results are left on the D: drive. The user can inspect them, and if they are satisfactory, they must be manually saved to the model file. This feature, similar to that used in SWOPSIM, insures that programs producing errors will not destroy model information. But the user is advised always to have backup copies of model files available just in case mistakes are made.

```
   DWOPSIM - Dynamic WOrld Policy SIMulation model building framework

   DWOPSIM is a set of DOS batch computer programs and SuperCalc 5
   spreadsheet templates that allow a user to assemble globally
   consistent data sets into time series models in spreadsheets.  Such
   models, which have all of the convenience and user access associated
   with spreadsheets, can be used for policy analysis, short term
   forecasting, and long term projections.  DWOPSIM models clear world
   trade markets through world price changes which are linked to product
   behavior in individual countries and regions.  The user is free to
   choose model structure and equation specification.

   Complete documentation of DWOPSIM programs and an illustrative model
   named TIME is found in:  Roningen, Vernon O., Documentation of the
   Dynamic World Policy Simulation (DWOPSIM) Model Building Framework,
   Staff Report No. AGES 9226, Econ. Res. Serv., U.S. Dept. Agr.,
   Oct. 1992 (Call 1-800-999-6779 or 1-703-834-0125 from outside the U.S.
   or Canada for information on ordering the DWOPSIM documentation).

   When you strike a key, default parameters for SuperCalc 5 will be set.

:DWOPSIM
ECHO OFF
CLS
TYPE C:\DWOPSIM\SCRN1
PAUSE
CLS
IF NOT EXIST SC.BAT GOTO C1
CALL SC C:\DWOPSIM\SETSC5
GOTO END
:C1
ECHO ERROR = You have not installed a DOS batch program SC.BAT which calls
ECHO         SuperCalc 5.  This must exist on the C: root directory and will
ECHO         look like the following example, assuming SuperCalc 5 is
ECHO         installed on the E:\CALC subdirectory.....................
ECHO
ECHO                          :SC.BAT
ECHO                          E:
ECHO                          CD \CALC
ECHO                          SC5 %1
ECHO                          CD \
ECHO                          C:
ECHO                          CLS
:END

:DWOPTIME
ECHO OFF
CLS
ECHO              DWOPSIM Program to create TSView-based data for model TIME
ECHO ------------------------------------------------------------------------
ECHO DWOPTIME      Program to make a complete set of balanced supply and
ECHO               demand spreadsheets for a selected PRoduct from a TSView
ECHO               data set for each CountrY in the model TIME, including a
ECHO               globally balanced RW (Rest-of-World bal. with DWOPBLRW).
ECHO REQUIREMENTS  A E:\TS subdirectory must contain the PRoduct TSVIEW files.
ECHO               A subtraction control file for TSView software must be on
ECHO               the model NAME subdirectory with the name NAMERW.SUB.
ECHO               An example of such a file for a 3 region world (US, EC,
ECHO               and RW) where RW = WD - US - EC might be:
ECHO               ------------------------------------------------------
ECHO               TEMP.TS
ECHO               TEMP.TS
ECHO               1
ECHO               RW                        1  3  5
ECHO                                  RW = WD-US-EC (Codes for TSView #s)
ECHO               ------------------------------------------------------
ECHO OUTPUT (D:)   CYPR spreadsheets for each CountrY in TIME plus the
ECHO               globally balanced RWPR spreadsheet.
ECHO ------------------------------------------------------------------------
ECHO COMMAND       DWOPTIME PR
ECHO ------------------------------------------------------------------------
rem  To convert this program to other models, 1) change TIME to new model
rem  name (marked by rem------ statements below), 2) change country
rem  numbers/codes in NAMERW.SUB and put on new model subdirectory
rem  (with new name), and 3) change country calls below to correspond to
rem  countries in new model (marked by rem******* statements below).
rem  NOTE: Steps 1) and 2) can be accomplished by invoking the program
rem        TIMEREPL.  Step 3 MUST be carried out manually.
IF FILE%1 == FILE  ECHO ERROR = You forgot PRoduct code; Enter: DWOPTIME PR
IF FILE%1 == FILE  GOTO END
rem Change TIME to name of new model-----------------------------------------
IF EXIST C:\TIME\TIMERW.SUB GOTO C1
ECHO ERROR = C:\TIME\TIMERW.SUB does not exist!  Create it first!
rem ------------------------------------------------------------------------
GOTO END
:C1
rem First set of country calling programs (change for a different model)*****
CALL DWOPGSUP US %1
CALL DWOPGSUP EC %1
IF NOT EXIST D:US%1.CAL GOTO C2
IF NOT EXIST D:EC%1.CAL GOTO C2
rem ************************************************************************
GOTO C3
:C2
ECHO A CountrY PRoduct (CYPR.CAL) file is missing on D:
GOTO END
:C3
rem Change TIME to name of new model-----------------------------------------
CALL DWOPRW TIME %1
rem ------------------------------------------------------------------------
IF EXIST D:RW%1.CAL GOTO C4
ECHO ERROR = The RW%1.CAL was not created on D:
```

```
GOTO END
:C4
rem  This section globally balances RW
ECHO {MACRO} >D:T.XQT
ECHO /LD:RW%1,R~ >>D:T.XQT
rem The RW balancing scheme in the spreadsheet DWOPBLRW.CAL will work as
rem long as quantity data in a model appears in the same columns as it
rem appears in the model TIME.  This arrangement of data follows that in
rem the TSVIEW data set which is the source of data for TIME.
ECHO /LC:\DWOPSIM\DWOPBLRW,A~ >>D:T.XQT
rem Second set of country calling programs**********************************
ECHO /LD:US%1,PU5:U65,AO5,+~ >>D:T.XQT
ECHO /LD:EC%1,PU5:U65,AO5,+~ >>D:T.XQT
ECHO /LD:RW%1,PU5:U65,AO5,+~ >>D:T.XQT
rem ************************************************************************
ECHO {CALC} >>D:T.XQT
ECHO /CAP5:AP65,D5,V~ >>D:T.XQT
ECHO /CAQ5:AQ65,E5,V~ >>D:T.XQT
ECHO /CAR5:AR65,F5,V~ >>D:T.XQT
ECHO /CAS5:AS65,G5,V~ >>D:T.XQT
ECHO /CAT5:AT65,I5,V~ >>D:T.XQT
ECHO {CALC} >>D:T.XQT
ECHO /BAO:AT~ >>D:T.XQT
ECHO =A1~ >>D:T.XQT
ECHO /SD:RW%1,OA~ >>D:T.XQT
ECHO {BEEP 2} >>D:T.XQT
ECHO /Q,Y~ >>D:T.XQT
CALL SC D:T
ERASE D:T.*
CLS
ECHO ...
ECHO The globally balanced files for %1 for the model are on D:.
ECHO ...
DIR D:/W
:END ECHO ON

:DWOPGSUP
ECHO OFF
CLS
ECHO               DWOPSIM Program
ECHO ------------------------------------------------------------------------
ECHO DWOPGSUP      Program to Get SUPply and demand data for a DWOPSIM
ECHO               selected CountrY and PRoduct from a TSView TS data set.
ECHO REQUIREMENTS  The PRoduct TS file is assumed to be on the E:\TS
ECHO               subdirectory and data is assumed to be in rows 5 through
ECHO               65 of the country/commodity spreadsheet.
ECHO               A template file DWOPTEMP.CAL must be on the C:\DWOPSIM
ECHO               subdirectory.
ECHO OUTPUT (D:)   The CYPR spreadsheet on D:.
ECHO ------------------------------------------------------------------------
ECHO COMMAND       DWOPGSUP CY PR
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE  ECHO ERROR = You forgot CountrY code; Enter: DWOPGSUP CY PR
IF FILE%1 == FILE  GOTO END
IF FILE%2 == FILE  ECHO ERROR = You forgot PRoduct code; Enter: DWOPGSUP CY PR
IF FILE%2 == FILE  GOTO END
IF EXIST E:\TS\%2.TS GOTO C1
ECHO ERROR = The time series file E:\TS\%2.TS does not exist!
GOTO END
:C1
IF EXIST C:\DWOPSIM\DWOPTEMP.CAL GOTO C2
ECHO ERROR = The template file DWOPTEMP.CAL does not exist on C:\DWOPSIM.
GOTO END
:C2
CLS
ECHO ...
ECHO The TS database is now being accessed and manipulated.
ECHO ...
IF EXIST D:?.XQT.CAL ERASE D:?.XQT.CAL
IF EXIST D:%1%2.CAL ERASE D:%1%2.CAL
IF EXIST D:T.CAL ERASE D:T.CAL
rem If the TS files are on another directory, change these statements.
COPY E:\TS\%2.TS D:
COPY E:\TS\TSSUBSET.EXE D:
COPY E:\TS\TS2PRN.EXE D:
ECHO %2 >D:T.BSS
ECHO T >>D:T.BSS
rem  The following statements are keyed to 143 region TSView database.
IF F%1 == FWD ECHO 1 >>D:T.BSS
IF F%1 == FFG ECHO 2 >>D:T.BSS
IF F%1 == FUS ECHO 3 >>D:T.BSS
IF F%1 == FCN ECHO 4 >>D:T.BSS
IF F%1 == FEC ECHO 5 >>D:T.BSS
IF F%1 == FWE ECHO 6 >>D:T.BSS
IF F%1 == FJP ECHO 7 >>D:T.BSS
IF F%1 == FAU ECHO 8 >>D:T.BSS
IF F%1 == FNZ ECHO 9 >>D:T.BSS
IF F%1 == FSF ECHO 10 >>D:T.BSS
IF F%1 == FEE ECHO 11 >>D:T.BSS
IF F%1 == FSV ECHO 12 >>D:T.BSS
IF F%1 == FCH ECHO 13 >>D:T.BSS
IF F%1 == FMX ECHO 14 >>D:T.BSS
IF F%1 == FCA ECHO 15 >>D:T.BSS
IF F%1 == FAR ECHO 16 >>D:T.BSS
IF F%1 == FBZ ECHO 17 >>D:T.BSS
IF F%1 == FVE ECHO 18 >>D:T.BSS
IF F%1 == FLA ECHO 19 >>D:T.BSS
IF F%1 == FNG ECHO 20 >>D:T.BSS
IF F%1 == FAF ECHO 21 >>D:T.BSS
IF F%1 == FEG ECHO 22 >>D:T.BSS
IF F%1 == FMP ECHO 23 >>D:T.BSS
IF F%1 == FMO ECHO 24 >>D:T.BSS
IF F%1 == FND ECHO 25 >>D:T.BSS
IF F%1 == FOS ECHO 26 >>D:T.BSS
```

```
IF F%1 == FDO ECHO 27 >>D:T.BSS
IF F%1 == FTH ECHO 28 >>D:T.BSS
IF F%1 == FML ECHO 29 >>D:T.BSS
IF F%1 == FPH ECHO 30 >>D:T.BSS
IF F%1 == FSA ECHO 31 >>D:T.BSS
IF F%1 == FSK ECHO 32 >>D:T.BSS
IF F%1 == FTW ECHO 33 >>D:T.BSS
IF F%1 == FEA ECHO 34 >>D:T.BSS
IF F%1 == FRW ECHO 35 >>D:T.BSS
IF F%1 == FDA ECHO 36 >>D:T.BSS
IF F%1 == FDE ECHO 37 >>D:T.BSS
IF F%1 == FBL ECHO 39 >>D:T.BSS
IF F%1 == FDN ECHO 39 >>D:T.BSS
IF F%1 == FFR ECHO 40 >>D:T.BSS
IF F%1 == FGC ECHO 41 >>D:T.BSS
IF F%1 == FIR ECHO 42 >>D:T.BSS
IF F%1 == FIT ECHO 43 >>D:T.BSS
IF F%1 == FNT ECHO 44 >>D:T.BSS
IF F%1 == FPT ECHO 45 >>D:T.BSS
IF F%1 == FSP ECHO 46 >>D:T.BSS
IF F%1 == FUK ECHO 47 >>D:T.BSS
IF F%1 == FWG ECHO 48 >>D:T.BSS
IF F%1 == FGD ECHO 49 >>D:T.BSS
IF F%1 == FAT ECHO 50 >>D:T.BSS
IF F%1 == FFN ECHO 51 >>D:T.BSS
IF F%1 == FNO ECHO 52 >>D:T.BSS
IF F%1 == FSW ECHO 53 >>D:T.BSS
IF F%1 == FSZ ECHO 54 >>D:T.BSS
IF F%1 == FOW ECHO 55 >>D:T.BSS
IF F%1 == FAB ECHO 56 >>D:T.BSS
IF F%1 == FBI ECHO 57 >>D:T.BSS
IF F%1 == FCZ ECHO 58 >>D:T.BSS
IF F%1 == FHU ECHO 59 >>D:T.BSS
IF F%1 == FPL ECHO 60 >>D:T.BSS
IF F%1 == FRM ECHO 61 >>D:T.BSS
IF F%1 == FYU ECHO 62 >>D:T.BSS
IF F%1 == FTK ECHO 63 >>D:T.BSS
IF F%1 == FNF ECHO 64 >>D:T.BSS
IF F%1 == FOM ECHO 65 >>D:T.BSS
IF F%1 == FR1 ECHO 66 >>D:T.BSS
IF F%1 == FBE ECHO 67 >>D:T.BSS
IF F%1 == FCR ECHO 68 >>D:T.BSS
IF F%1 == FES ECHO 69 >>D:T.BSS
IF F%1 == FGT ECHO 70 >>D:T.BSS
IF F%1 == FHO ECHO 71 >>D:T.BSS
IF F%1 == FNI ECHO 72 >>D:T.BSS
IF F%1 == FPA ECHO 73 >>D:T.BSS
IF F%1 == FBH ECHO 74 >>D:T.BSS
IF F%1 == FBA ECHO 75 >>D:T.BSS
IF F%1 == FBD ECHO 76 >>D:T.BSS
IF F%1 == FCU ECHO 77 >>D:T.BSS
IF F%1 == FDR ECHO 78 >>D:T.BSS
IF F%1 == FGU ECHO 79 >>D:T.BSS
IF F%1 == FHA ECHO 80 >>D:T.BSS
IF F%1 == FJM ECHO 81 >>D:T.BSS
IF F%1 == FMA ECHO 82 >>D:T.BSS
IF F%1 == FNN ECHO 83 >>D:T.BSS
IF F%1 == FSC ECHO 84 >>D:T.BSS
IF F%1 == FST ECHO 85 >>D:T.BSS
IF F%1 == FTT ECHO 86 >>D:T.BSS
IF F%1 == FBO ECHO 87 >>D:T.BSS
IF F%1 == FCL ECHO 88 >>D:T.BSS
IF F%1 == FCO ECHO 89 >>D:T.BSS
IF F%1 == FED ECHO 90 >>D:T.BSS
IF F%1 == FGY ECHO 91 >>D:T.BSS
IF F%1 == FPR ECHO 92 >>D:T.BSS
IF F%1 == FPE ECHO 93 >>D:T.BSS
IF F%1 == FSU ECHO 94 >>D:T.BSS
IF F%1 == FUR ECHO 95 >>D:T.BSS
IF F%1 == FR2 ECHO 96 >>D:T.BSS
IF F%1 == FMN ECHO 97 >>D:T.BSS
IF F%1 == FNK ECHO 98 >>D:T.BSS
IF F%1 == FBR ECHO 99 >>D:T.BSS
IF F%1 == FBU ECHO 100 >>D:T.BSS
IF F%1 == FKR ECHO 101 >>D:T.BSS
IF F%1 == FLO ECHO 102 >>D:T.BSS
IF F%1 == FSN ECHO 103 >>D:T.BSS
IF F%1 == FVT ECHO 104 >>D:T.BSS
IF F%1 == FAH ECHO 105 >>D:T.BSS
IF F%1 == FBG ECHO 106 >>D:T.BSS
IF F%1 == FPK ECHO 107 >>D:T.BSS
IF F%1 == FSL ECHO 108 >>D:T.BSS
IF F%1 == FOA ECHO 109 >>D:T.BSS
IF F%1 == FR3 ECHO 110 >>D:T.BSS
IF F%1 == FOE ECHO 111 >>D:T.BSS
IF F%1 == FAS ECHO 112 >>D:T.BSS
IF F%1 == FAL ECHO 113 >>D:T.BSS
IF F%1 == FLY ECHO 114 >>D:T.BSS
IF F%1 == FMC ECHO 115 >>D:T.BSS
IF F%1 == FTN ECHO 116 >>D:T.BSS
IF F%1 == FCM ECHO 117 >>D:T.BSS
IF F%1 == FGH ECHO 118 >>D:T.BSS
IF F%1 == FGN ECHO 119 >>D:T.BSS
IF F%1 == FIC ECHO 120 >>D:T.BSS
IF F%1 == FLB ECHO 121 >>D:T.BSS
IF F%1 == FSG ECHO 122 >>D:T.BSS
IF F%1 == FZR ECHO 123 >>D:T.BSS
IF F%1 == FAW ECHO 124 >>D:T.BSS
IF F%1 == FCF ECHO 125 >>D:T.BSS
IF F%1 == FEP ECHO 126 >>D:T.BSS
IF F%1 == FKY ECHO 127 >>D:T.BSS
IF F%1 == FSM ECHO 128 >>D:T.BSS
IF F%1 == FSD ECHO 129 >>D:T.BSS
IF F%1 == FTZ ECHO 130 >>D:T.BSS
```

```
IF F%1 == FUG ECHO 131 >>D:T.BSS
IF F%1 == FAE ECHO 132 >>D:T.BSS
IF F%1 == FAG ECHO 133 >>D:T.BSS
IF F%1 == FBT ECHO 134 >>D:T.BSS
IF F%1 == FLH ECHO 135 >>D:T.BSS
IF F%1 == FMG ECHO 136 >>D:T.BSS
IF F%1 == FMW ECHO 137 >>D:T.BSS
IF F%1 == FMZ ECHO 138 >>D:T.BSS
IF F%1 == FWZ ECHO 139 >>D:T.BSS
IF F%1 == FZA ECHO 140 >>D:T.BSS
IF F%1 == FZB ECHO 141 >>D:T.BSS
IF F%1 == FOF ECHO 142 >>D:T.BSS
IF F%1 == FR4 ECHO 143 >>D:T.BSS
D:
TSSUBSET T.BSS
TS2PRN T
C:
CLS
ECHO ...
ECHO Now the ASCII data from the TS system will be entered into %1%2.CAL.
ECHO ...
ECHO .
ECHO ...
ECHO A DWOPSIM program prepared by Vernon Oley Roningen
ECHO ...
ERASE D:TSSUBSET.EXE
ERASE D:TS2PRN.EXE
ERASE D:T.TS
ERASE D:%2.TS
ECHO {MACRO} >D:T.XQT
ECHO {PROMPT "%1%2.CAL will be on D: drive when the program is finished."}
>>D:T.XQT
ECHO {STATUS "Preparing data on %2 for %1"} >>D:T.XQT
ECHO {PANELOFF} >>D:T.XQT
ECHO {WINDOWSOFF} >>D:T.XQT
ECHO //INT~A~ >>D:T.XQT
ECHO /MR4~1~/MR4~2~ >>D:T.XQT
ECHO /CA2,A67~ >>D:T.XQT
ECHO /CC2,A68~ >>D:T.XQT
ECHO /SD:T~A~/Z,Y~ >>D:T.XQT
ECHO /LC:\DWOPSIM\DWOPTEMP.CAL,R~ >>D:T.XQT
ECHO /LD:T,A~ >>D:T.XQT
ECHO /FEA67:A68,TL~ >>D:T.XQT
ECHO /PA67:A68~ >>D:T.XQT
ECHO /BC2~ >>D:T.XQT
ECHO {LETC A1,%1%2} >>D:T.XQT
ECHO /PA1~ >>D:T.XQT
rem If years are in row other than 5 to 65, change the following statements.
FOR %%C IN (5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22) DO CALL DWOPPTNA
%%C
FOR %%C IN (23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38) DO CALL DWOPPTNA
%%C
FOR %%C IN (39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54) DO CALL DWOPPTNA %%C
FOR %%C IN (55 56 57 58 59 60 61 62 63 64 65) DO CALL DWOPPTNA %%C
ECHO {CALC} >>D:T.XQT
ECHO /UF5~{LET F5,IF((U5+E5)>=0,U5+E5,F5)}/UE5~{LET E5,IF((F5-U5)>=0,F5-U5,E5)}
>>D:T.XQT
ECHO /UF6~{LET F6,IF((U6+E6)>=0,U6+E6,F6)}/UE6~{LET E6,IF((F6-U6)>=0,F6-U6,E6)}
>>D:T.XQT
ECHO /UF7~{LET F7,IF((U7+E7)>=0,U7+E7,F7)}/UE7~{LET E7,IF((F7-U7)>=0,F7-U7,E7)}
>>D:T.XQT
ECHO /UF8~{LET F8,IF((U8+E8)>=0,U8+E8,F8)}/UE8~{LET E8,IF((F8-U8)>=0,F8-U8,E8)}
>>D:T.XQT
ECHO /UF9~{LET F9,IF((U9+E9)>=0,U9+E9,F9)}/UE9~{LET E9,IF((F9-U9)>=0,F9-U9,E9)}
>>D:T.XQT
ECHO   /UF10~{LET   F10,IF((U10+E10)>=0,U10+E10,F10)}/UE10~{LET
E10,IF((F10-U10)>=0,F10-U10,E10)} >>D:T.XQT
ECHO   /UF11~{LET   F11,IF((U11+E11)>=0,U11+E11,F11)}/UE11~{LET
E11,IF((F11-U11)>=0,F11-U11,E11)} >>D:T.XQT
ECHO   /UF12~{LET   F12,IF((U12+E12)>=0,U12+E12,F12)}/UE12~{LET
E12,IF((F12-U12)>=0,F12-U12,E12)} >>D:T.XQT
ECHO   /UF13~{LET   F13,IF((U13+E13)>=0,U13+E13,F13)}/UE13~{LET
E13,IF((F13-U13)>=0,F13-U13,E13)} >>D:T.XQT
ECHO   /UF14~{LET   F14,IF((U14+E14)>=0,U14+E14,F14)}/UE14~{LET
E14,IF((F14-U14)>=0,F14-U14,E14)} >>D:T.XQT
ECHO   /UF15~{LET   F15,IF((U15+E15)>=0,U15+E15,F15)}/UE15~{LET
E15,IF((F15-U15)>=0,F15-U15,E15)} >>D:T.XQT
ECHO   /UF16~{LET   F16,IF((U16+E16)>=0,U16+E16,F16)}/UE16~{LET
E16,IF((F16-U16)>=0,F16-U16,E16)} >>D:T.XQT
ECHO   /UF17~{LET   F17,IF((U17+E17)>=0,U17+E17,F17)}/UE17~{LET
E17,IF((F17-U17)>=0,F17-U17,E17)} >>D:T.XQT
ECHO   /UF18~{LET   F18,IF((U18+E18)>=0,U18+E18,F18)}/UE18~{LET
E18,IF((F18-U18)>=0,F18-U18,E18)} >>D:T.XQT
ECHO   /UF19~{LET   F19,IF((U19+E19)>=0,U19+E19,F19)}/UE19~{LET
E19,IF((F19-U19)>=0,F19-U19,E19)} >>D:T.XQT
ECHO   /UF20~{LET   F20,IF((U20+E20)>=0,U20+E20,F20)}/UE20~{LET
E20,IF((F20-U20)>=0,F20-U20,E20)} >>D:T.XQT
ECHO   /UF21~{LET   F21,IF((U21+E21)>=0,U21+E21,F21)}/UE21~{LET
E21,IF((F21-U21)>=0,F21-U21,E21)} >>D:T.XQT
ECHO   /UF22~{LET   F22,IF((U22+E22)>=0,U22+E22,F22)}/UE22~{LET
E22,IF((F22-U22)>=0,F22-U22,E22)} >>D:T.XQT
ECHO   /UF23~{LET   F23,IF((U23+E23)>=0,U23+E23,F23)}/UE23~{LET
E23,IF((F23-U23)>=0,F23-U23,E23)} >>D:T.XQT
ECHO   /UF24~{LET   F24,IF((U24+E24)>=0,U24+E24,F24)}/UE24~{LET
E24,IF((F24-U24)>=0,F24-U24,E24)} >>D:T.XQT
ECHO   /UF25~{LET   F25,IF((U25+E25)>=0,U25+E25,F25)}/UE25~{LET
E25,IF((F25-U25)>=0,F25-U25,E25)} >>D:T.XQT
ECHO   /UF26~{LET   F26,IF((U26+E26)>=0,U26+E26,F26)}/UE26~{LET
E26,IF((F26-U26)>=0,F26-U26,E26)} >>D:T.XQT
ECHO   /UF27~{LET   F27,IF((U27+E27)>=0,U27+E27,F27)}/UE27~{LET
E27,IF((F27-U27)>=0,F27-U27,E27)} >>D:T.XQT
ECHO   /UF28~{LET   F28,IF((U28+E28)>=0,U28+E28,F28)}/UE28~{LET
E28,IF((F28-U28)>=0,F28-U28,E28)} >>D:T.XQT
ECHO   /UF29~{LET   F29,IF((U29+E29)>=0,U29+E29,F29)}/UE29~{LET
```

```
E29,IF((F29-U29)>=0,F29-U29,E29)) >>D:T.XQT
ECHO   /UF30~{LET   F30,IF((U30+E30)>=0,U30+E30,F30))/UE30~{LET
E30,IF((F30-U30)>=0,F30-U30,E30)) >>D:T.XQT
ECHO   /UF31~{LET   F31,IF((U31+E31)>=0,U31+E31,F31))/UE31~{LET
E31,IF((F31-U31)>=0,F31-U31,E31)) >>D:T.XQT
ECHO   /UF32~{LET   F32,IF((U32+E32)>=0,U32+E32,F32))/UE32~{LET
E32,IF((F32-U32)>=0,F32-U32,E32)) >>D:T.XQT
ECHO   /UF33~{LET   F33,IF((U33+E33)>=0,U33+E33,F33))/UE33~{LET
E33,IF((F33-U33)>=0,F33-U33,E33)) >>D:T.XQT
ECHO   /UF34~{LET   F34,IF((U34+E34)>=0,U34+E34,F34))/UE34~{LET
E34,IF((F34-U34)>=0,F34-U34,E34)) >>D:T.XQT
ECHO   /UF35~{LET   F35,IF((U35+E35)>=0,U35+E35,F35))/UE35~{LET
E35,IF((F35-U35)>=0,F35-U35,E35)) >>D:T.XQT
ECHO   /UF36~{LET   F36,IF((U36+E36)>=0,U36+E36,F36))/UE36~{LET
E36,IF((F36-U36)>=0,F36-U36,E36)) >>D:T.XQT
ECHO   /UF37~{LET   F37,IF((U37+E37)>=0,U37+E37,F37))/UE37~{LET
E37,IF((F37-U37)>=0,F37-U37,E37)) >>D:T.XQT
ECHO   /UF38~{LET   F38,IF((U38+E38)>=0,U38+E38,F38))/UE38~{LET
E38,IF((F38-U38)>=0,F38-U38,E38)) >>D:T.XQT
ECHO   /UF39~{LET   F39,IF((U39+E39)>=0,U39+E39,F39))/UE39~{LET
E39,IF((F39-U39)>=0,F39-U39,E39)) >>D:T.XQT
ECHO   /UF40~{LET   F40,IF((U40+E40)>=0,U40+E40,F40))/UE40~{LET
E40,IF((F40-U40)>=0,F40-U40,E40)) >>D:T.XQT
ECHO   /UF41~{LET   F41,IF((U41+E41)>=0,U41+E41,F41))/UE41~{LET
E41,IF((F41-U41)>=0,F41-U41,E41)) >>D:T.XQT
ECHO   /UF42~{LET   F42,IF((U42+E42)>=0,U42+E42,F42))/UE42~{LET
E42,IF((F42-U42)>=0,F42-U42,E42)) >>D:T.XQT
ECHO   /UF43~{LET   F43,IF((U43+E43)>=0,U43+E43,F43))/UE43~{LET
E43,IF((F43-U43)>=0,F43-U43,E43)) >>D:T.XQT
ECHO   /UF44~{LET   F44,IF((U44+E44)>=0,U44+E44,F44))/UE44~{LET
E44,IF((F44-U44)>=0,F44-U44,E44)) >>D:T.XQT
ECHO   /UF45~{LET   F45,IF((U45+E45)>=0,U45+E45,F45))/UE45~{LET
E45,IF((F45-U45)>=0,F45-U45,E45)) >>D:T.XQT
ECHO   /UF46~{LET   F46,IF((U46+E46)>=0,U46+E46,F46))/UE46~{LET
E46,IF((F46-U46)>=0,F46-U46,E46)) >>D:T.XQT
ECHO   /UF47~{LET   F47,IF((U47+E47)>=0,U47+E47,F47))/UE47~{LET
E47,IF((F47-U47)>=0,F47-U47,E47)) >>D:T.XQT
ECHO   /UF48~{LET   F48,IF((U48+E48)>=0,U48+E48,F48))/UE48~{LET
E48,IF((F48-U48)>=0,F48-U48,E48)) >>D:T.XQT
ECHO   /UF49~{LET   F49,IF((U49+E49)>=0,U49+E49,F49))/UE49~{LET
E49,IF((F49-U49)>=0,F49-U49,E49)) >>D:T.XQT
ECHO   /UF50~{LET   F50,IF((U50+E50)>=0,U50+E50,F50))/UE50~{LET
E50,IF((F50-U50)>=0,F50-U50,E50)) >>D:T.XQT
ECHO   /UF51~{LET   F51,IF((U51+E51)>=0,U51+E51,F51))/UE51~{LET
E51,IF((F51-U51)>=0,F51-U51,E51)) >>D:T.XQT
ECHO   /UF52~{LET   F52,IF((U52+E52)>=0,U52+E52,F52))/UE52~{LET
E52,IF((F52-U52)>=0,F52-U52,E52)) >>D:T.XQT
ECHO   /UF53~{LET   F53,IF((U53+E53)>=0,U53+E53,F53))/UE53~{LET
E53,IF((F53-U53)>=0,F53-U53,E53)) >>D:T.XQT
ECHO   /UF54~{LET   F54,IF((U54+E54)>=0,U54+E54,F54))/UE54~{LET
E54,IF((F54-U54)>=0,F54-U54,E54)) >>D:T.XQT
ECHO   /UF55~{LET   F55,IF((U55+E55)>=0,U55+E55,F55))/UE55~{LET
E55,IF((F55-U55)>=0,F55-U55,E55)) >>D:T.XQT
ECHO   /UF56~{LET   F56,IF((U56+E56)>=0,U56+E56,F56))/UE56~{LET
E56,IF((F56-U56)>=0,F56-U56,E56)) >>D:T.XQT
ECHO   /UF57~{LET   F57,IF((U57+E57)>=0,U57+E57,F57))/UE57~{LET
E57,IF((F57-U57)>=0,F57-U57,E57)) >>D:T.XQT
ECHO   /UF58~{LET   F58,IF((U58+E58)>=0,U58+E58,F58))/UE58~{LET
E58,IF((F58-U58)>=0,F58-U58,E58)) >>D:T.XQT
ECHO   /UF59~{LET   F59,IF((U59+E59)>=0,U59+E59,F59))/UE59~{LET
E59,IF((F59-U59)>=0,F59-U59,E59)) >>D:T.XQT
ECHO   /UF60~{LET   F60,IF((U60+E60)>=0,U60+E60,F60))/UE60~{LET
E60,IF((F60-U60)>=0,F60-U60,E60)) >>D:T.XQT
ECHO   /UF61~{LET   F61,IF((U61+E61)>=0,U61+E61,F61))/UE61~{LET
E61,IF((F61-U61)>=0,F61-U61,E61)) >>D:T.XQT
ECHO   /UF62~{LET   F62,IF((U62+E62)>=0,U62+E62,F62))/UE62~{LET
E62,IF((F62-U62)>=0,F62-U62,E62)) >>D:T.XQT
ECHO   /UF63~{LET   F63,IF((U63+E63)>=0,U63+E63,F63))/UE63~{LET
E63,IF((F63-U63)>=0,F63-U63,E63)) >>D:T.XQT
ECHO   /UF64~{LET   F64,IF((U64+E64)>=0,U64+E64,F64))/UE64~{LET
E64,IF((F64-U64)>=0,F64-U64,E64)) >>D:T.XQT
ECHO   /UF65~{LET   F65,IF((U65+E65)>=0,U65+E65,F65))/UE65~{LET
E65,IF((F65-U65)>=0,F65-U65,E65)) >>D:T.XQT
ECHO /SD:%1%2,A~ >>D:T.XQT
ECHO (BEEP 1) >>D:T.XQT
ECHO /Q,Y~ >>D:T.XQT
CALL SC D:T
ERASE D:T.*
DIR D:/W
:END ECHO ON
:DWOPRW
ECHO OFF
CLS
DWOPSIM Program
---------------------------------------------------------------------------
ECHO DWOPRW      Program to make a balanced RW (Rest-of-World) supply and
ECHO             demand spreadsheet for a selected Product from
ECHO             as TSView TS data set - with the name RWPR.
ECHO REQUIREMENTS A E:\TS subdirectory must contain the Product TS file.
ECHO             A subtraction control file for TSView software must be
ECHO             on the model NAME subdirectory with the name NAMERW.SUB.
ECHO             For a 3 region world (US, EC, and RW) an example of such
ECHO             a file where RW = WD - US - EC might be:
ECHO             ----------------------------------------
ECHO             TEMP.TS
ECHO             TEMP.TS
ECHO             1
ECHO             RW                    1  3  5
ECHO                                   RW = WD-US-EC (Codes for TS #s above)
ECHO             ----------------------------------------
ECHO OUTPUT (D:) The globally balanced RWPR spreadsheet on D:.
ECHO ---------------------------------------------------------------------------
ECHO COMMAND     DWOPRW NAME PR
```

```
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE  ECHO ERROR = You forgot model NAME; Enter: DWOPRW NAME PR
IF FILE%1 == FILE  GOTO END
IF FILE%2 == FILE  ECHO ERROR = You forgot PRoduct code; Enter: DWOPRW NAME PR
IF FILE%2 == FILE  GOTO END
REM If the TS system is not on the E: drive, change the letter!
IF EXIST E:\TS\%2.TS GOTO C1
ECHO ERROR = E:\TS\%2.TS does not exist!
GOTO END
:C1
IF EXIST C:\%1\%1RW.SUB GOTO C2
ECHO ERROR = C:\%1\%1RW.SUB does not exist!  Create it first!
GOTO END
:C2
IF EXIST C:\DWOPSIM\DWOPTEMP.CAL GOTO C3
ECHO ERROR = The template file DWOPTEMP.CAL does not exist on C:\DWOPSIM.
:C3
CLS
ECHO ...
ECHO The TS database is now being accessed and manipulated.
ECHO ...
IF EXIST D:?.XQT.CAL ERASE D:?.XQT.CAL
IF EXIST D:%2.CAL ERASE D:%2.CAL
IF EXIST D:%1RW.SUB ERASE D:%1RW.CAL
IF EXIST D:RW%2.CAL ERASE D:RW%2.CAL
REM If the TS system is not on the E: drive, change the letter!
COPY E:\TS\%2.TS D:TEMP.TS
COPY E:\TS\TSSUB.EXE D:
COPY E:\TS\TS2PRN.EXE D:
COPY E:\TS\TSSUBSET.EXE D:
COPY C:\%1\%1RW.SUB D:
ECHO TEMP >D:T.BSS
ECHO T >>D:T.BSS
ECHO 144 >>D:T.BSS
D:
TSSUB %1RW
TSSUBSET T.BSS
TS2PRN T
C:
CLS
ECHO ...
ECHO Now the ASCII data from the TS system will be entered into RW%2.CAL.
ECHO ...
ECHO .
ECHO ...
ECHO A DWOPSIM program prepared by Vernon Oley Roningen
ECHO ...
ERASE D:TSSUB.EXE
ERASE D:TSSUBSET.EXE
ERASE D:TS2PRN.EXE
ERASE D:%1RW.SUB
ERASE D:T.TS
ERASE D:TEMP.*
ECHO {MACRO} >D:T.XQT
ECHO {PROMPT "RW%2.CAL for model %1 will be on the D: drive when the program is
finished."} >>D:T.XQT
ECHO {STATUS "Preparing data on %2 for RW"} >>D:T.XQT
ECHO {PANELOFF} >>D:T.XQT
ECHO {WINDOWSOFF} >>D:T.XQT
ECHO //INT~A~ >>D:T.XQT
ECHO /MR4~1~/MR4~2~ >>D:T.XQT
ECHO /CA2,A67~ >>D:T.XQT
ECHO /CC2,A68~ >>D:T.XQT
ECHO /SD:T~A~/Z,Y~ >>D:T.XQT
ECHO /LC:\DWOPSIM\DWOPTEMP.CAL,R~  >>D:T.XQT
ECHO /LD:T,A~ >>D:T.XQT
ECHO /FEA67:A68,TL~ >>D:T.XQT
ECHO /PA67:A68~ >>D:T.XQT
ECHO /BC2~ >>D:T.XQT
ECHO {LETC A1,RW%2} >>D:T.XQT
ECHO /PA1~ >>D:T.XQT
rem All of the following statements assume time series data is put in
rem spreadsheet rows 5 - 65 (corresponding to TSView data running from
rem 1960 to 1990.  If the number of years changes, these numbers must be
rem changed accordingly.
FOR %%C IN (5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22) DO CALL DWOPPTNA
%%C
FOR %%C IN (23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38) DO CALL DWOPPTNA %%C
FOR %%C IN (39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54) DO CALL DWOPPTNA %%C
FOR %%C IN (55 56 57 58 59 60 61 62 63 64 65) DO CALL DWOPPTNA %%C
ECHO {CALC} >>D:T.XQT
ECHO /UF5~{LET F5,IF((U5+E5)>=0,U5+E5,F5)}/UE5~{LET E5,IF((F5-U5)>=0,F5-U5,E5)}
>>D:T.XQT
ECHO /UF6~{LET F6,IF((U6+E6)>=0,U6+E6,F6)}/UE6~{LET E6,IF((F6-U6)>=0,F6-U6,E6)}
>>D:T.XQT
ECHO /UF7~{LET F7,IF((U7+E7)>=0,U7+E7,F7)}/UE7~{LET E7,IF((F7-U7)>=0,F7-U7,E7)}
>>D:T.XQT
ECHO /UF8~{LET F8,IF((U8+E8)>=0,U8+E8,F8)}/UE8~{LET E8,IF((F8-U8)>=0,F8-U8,E8)}
>>D:T.XQT
ECHO /UF9~{LET F9,IF((U9+E9)>=0,U9+E9,F9)}/UE9~{LET E9,IF((F9-U9)>=0,F9-U9,E9)}
>>D:T.XQT
ECHO   /UF10~{LET   F10,IF((U10+E10)>=0,U10+E10,F10)}/UE10~{LET
E10,IF((F10-U10)>=0,F10-U10,E10)} >>D:T.XQT
ECHO   /UF11~{LET   F11,IF((U11+E11)>=0,U11+E11,F11)}/UE11~{LET
E11,IF((F11-U11)>=0,F11-U11,E11)} >>D:T.XQT
ECHO   /UF12~{LET   F12,IF((U12+E12)>=0,U12+E12,F12)}/UE12~{LET
E12,IF((F12-U12)>=0,F12-U12,E12)} >>D:T.XQT
ECHO   /UF13~{LET   F13,IF((U13+E13)>=0,U13+E13,F13)}/UE13~{LET
E13,IF((F13-U13)>=0,F13-U13,E13)} >>D:T.XQT
ECHO   /UF14~{LET   F14,IF((U14+E14)>=0,U14+E14,F14)}/UE14~{LET
E14,IF((F14-U14)>=0,F14-U14,E14)} >>D:T.XQT
ECHO   /UF15~{LET   F15,IF((U15+E15)>=0,U15+E15,F15)}/UE15~{LET
E15,IF((F15-U15)>=0,F15-U15,E15)} >>D:T.XQT
ECHO   /UF16~{LET   F16,IF((U16+E16)>=0,U16+E16,F16)}/UE16~{LET
```

```
E16,IF((F16-U16)>=0,F16-U16,E16)} >>D:T.XQT
ECHO   /UF17~{LET   F17,IF((U17+E17)>=0,U17+E17,F17)}/UE17~{LET
E17,IF((F17-U17)>=0,F17-U17,E17)} >>D:T.XQT
ECHO   /UF18~{LET   F18,IF((U18+E18)>=0,U18+E18,F18)}/UE18~{LET
E18,IF((F18-U18)>=0,F18-U18,E18)} >>D:T.XQT
ECHO   /UF19~{LET   F19,IF((U19+E19)>=0,U19+E19,F19)}/UE19~{LET
E19,IF((F19-U19)>=0,F19-U19,E19)} >>D:T.XQT
ECHO   /UF20~{LET   F20,IF((U20+E20)>=0,U20+E20,F20)}/UE20~{LET
E20,IF((F20-U20)>=0,F20-U20,E20)} >>D:T.XQT
ECHO   /UF21~{LET   F21,IF((U21+E21)>=0,U21+E21,F21)}/UE21~{LET
E21,IF((F21-U21)>=0,F21-U21,E21)} >>D:T.XQT
ECHO   /UF22~{LET   F22,IF((U22+E22)>=0,U22+E22,F22)}/UE22~{LET
E22,IF((F22-U22)>=0,F22-U22,E22)} >>D:T.XQT
ECHO   /UF23~{LET   F23,IF((U23+E23)>=0,U23+E23,F23)}/UE23~{LET
E23,IF((F23-U23)>=0,F23-U23,E23)} >>D:T.XQT
ECHO   /UF24~{LET   F24,IF((U24+E24)>=0,U24+E24,F24)}/UE24~{LET
E24,IF((F24-U24)>=0,F24-U24,E24)} >>D:T.XQT
ECHO   /UF25~{LET   F25,IF((U25+E25)>=0,U25+E25,F25)}/UE25~{LET
E25,IF((F25-U25)>=0,F25-U25,E25)} >>D:T.XQT
ECHO   /UF26~{LET   F26,IF((U26+E26)>=0,U26+E26,F26)}/UE26~{LET
E26,IF((F26-U26)>=0,F26-U26,E26)} >>D:T.XQT
ECHO   /UF27~{LET   F27,IF((U27+E27)>=0,U27+E27,F27)}/UE27~{LET
E27,IF((F27-U27)>=0,F27-U27,E27)} >>D:T.XQT
ECHO   /UF28~{LET   F28,IF((U28+E28)>=0,U28+E28,F28)}/UE28~{LET
E28,IF((F28-U28)>=0,F28-U28,E28)} >>D:T.XQT
ECHO   /UF29~{LET   F29,IF((U29+E29)>=0,U29+E29,F29)}/UE29~{LET
E29,IF((F29-U29)>=0,F29-U29,E29)} >>D:T.XQT
ECHO   /UF30~{LET   F30,IF((U30+E30)>=0,U30+E30,F30)}/UE30~{LET
E30,IF((F30-U30)>=0,F30-U30,E30)} >>D:T.XQT
ECHO   /UF31~{LET   F31,IF((U31+E31)>=0,U31+E31,F31)}/UE31~{LET
E31,IF((F31-U31)>=0,F31-U31,E31)} >>D:T.XQT
ECHO   /UF32~{LET   F32,IF((U32+E32)>=0,U32+E32,F32)}/UE32~{LET
E32,IF((F32-U32)>=0,F32-U32,E32)} >>D:T.XQT
ECHO   /UF33~{LET   F33,IF((U33+E33)>=0,U33+E33,F33)}/UE33~{LET
E33,IF((F33-U33)>=0,F33-U33,E33)} >>D:T.XQT
ECHO   /UF34~{LET   F34,IF((U34+E34)>=0,U34+E34,F34)}/UE34~{LET
E34,IF((F34-U34)>=0,F34-U34,E34)} >>D:T.XQT
ECHO   /UF35~{LET   F35,IF((U35+E35)>=0,U35+E35,F35)}/UE35~{LET
E35,IF((F35-U35)>=0,F35-U35,E35)} >>D:T.XQT
ECHO   /UF36~{LET   F36,IF((U36+E36)>=0,U36+E36,F36)}/UE36~{LET
E36,IF((F36-U36)>=0,F36-U36,E36)} >>D:T.XQT
ECHO   /UF37~{LET   F37,IF((U37+E37)>=0,U37+E37,F37)}/UE37~{LET
E37,IF((F37-U37)>=0,F37-U37,E37)} >>D:T.XQT
ECHO   /UF38~{LET   F38,IF((U38+E38)>=0,U38+E38,F38)}/UE38~{LET
E38,IF((F38-U38)>=0,F38-U38,E38)} >>D:T.XQT
ECHO   /UF39~{LET   F39,IF((U39+E39)>=0,U39+E39,F39)}/UE39~{LET
E39,IF((F39-U39)>=0,F39-U39,E39)} >>D:T.XQT
ECHO   /UF40~{LET   F40,IF((U40+E40)>=0,U40+E40,F40)}/UE40~{LET
E40,IF((F40-U40)>=0,F40-U40,E40)} >>D:T.XQT
ECHO   /UF41~{LET   F41,IF((U41+E41)>=0,U41+E41,F41)}/UE41~{LET
E41,IF((F41-U41)>=0,F41-U41,E41)} >>D:T.XQT
ECHO   /UF42~{LET   F42,IF((U42+E42)>=0,U42+E42,F42)}/UE42~{LET
E42,IF((F42-U42)>=0,F42-U42,E42)} >>D:T.XQT
ECHO   /UF43~{LET   F43,IF((U43+E43)>=0,U43+E43,F43)}/UE43~{LET
E43,IF((F43-U43)>=0,F43-U43,E43)} >>D:T.XQT
ECHO   /UF44~{LET   F44,IF((U44+E44)>=0,U44+E44,F44)}/UE44~{LET
E44,IF((F44-U44)>=0,F44-U44,E44)} >>D:T.XQT
ECHO   /UF45~{LET   F45,IF((U45+E45)>=0,U45+E45,F45)}/UE45~{LET
E45,IF((F45-U45)>=0,F45-U45,E45)} >>D:T.XQT
ECHO   /UF46~{LET   F46,IF((U46+E46)>=0,U46+E46,F46)}/UE46~{LET
E46,IF((F46-U46)>=0,F46-U46,E46)} >>D:T.XQT
ECHO   /UF47~{LET   F47,IF((U47+E47)>=0,U47+E47,F47)}/UE47~{LET
E47,IF((F47-U47)>=0,F47-U47,E47)} >>D:T.XQT
ECHO   /UF48~{LET   F48,IF((U48+E48)>=0,U48+E48,F48)}/UE48~{LET
E48,IF((F48-U48)>=0,F48-U48,E48)} >>D:T.XQT
ECHO   /UF49~{LET   F49,IF((U49+E49)>=0,U49+E49,F49)}/UE49~{LET
E49,IF((F49-U49)>=0,F49-U49,E49)} >>D:T.XQT
ECHO   /UF50~{LET   F50,IF((U50+E50)>=0,U50+E50,F50)}/UE50~{LET
E50,IF((F50-U50)>=0,F50-U50,E50)} >>D:T.XQT
ECHO   /UF51~{LET   F51,IF((U51+E51)>=0,U51+E51,F51)}/UE51~{LET
E51,IF((F51-U51)>=0,F51-U51,E51)} >>D:T.XQT
ECHO   /UF52~{LET   F52,IF((U52+E52)>=0,U52+E52,F52)}/UE52~{LET
E52,IF((F52-U52)>=0,F52-U52,E52)} >>D:T.XQT
ECHO   /UF53~{LET   F53,IF((U53+E53)>=0,U53+E53,F53)}/UE53~{LET
E53,IF((F53-U53)>=0,F53-U53,E53)} >>D:T.XQT
ECHO   /UF54~{LET   F54,IF((U54+E54)>=0,U54+E54,F54)}/UE54~{LET
E54,IF((F54-U54)>=0,F54-U54,E54)} >>D:T.XQT
ECHO   /UF55~{LET   F55,IF((U55+E55)>=0,U55+E55,F55)}/UE55~{LET
E55,IF((F55-U55)>=0,F55-U55,E55)} >>D:T.XQT
ECHO   /UF56~{LET   F56,IF((U56+E56)>=0,U56+E56,F56)}/UE56~{LET
E56,IF((F56-U56)>=0,F56-U56,E56)} >>D:T.XQT
ECHO   /UF57~{LET   F57,IF((U57+E57)>=0,U57+E57,F57)}/UE57~{LET
E57,IF((F57-U57)>=0,F57-U57,E57)} >>D:T.XQT
ECHO   /UF58~{LET   F58,IF((U58+E58)>=0,U58+E58,F58)}/UE58~{LET
E58,IF((F58-U58)>=0,F58-U58,E58)} >>D:T.XQT
ECHO   /UF59~{LET   F59,IF((U59+E59)>=0,U59+E59,F59)}/UE59~{LET
E59,IF((F59-U59)>=0,F59-U59,E59)} >>D:T.XQT
ECHO   /UF60~{LET   F60,IF((U60+E60)>=0,U60+E60,F60)}/UE60~{LET
E60,IF((F60-U60)>=0,F60-U60,E60)} >>D:T.XQT
ECHO   /UF61~{LET   F61,IF((U61+E61)>=0,U61+E61,F61)}/UE61~{LET
E61,IF((F61-U61)>=0,F61-U61,E61)} >>D:T.XQT
ECHO   /UF62~{LET   F62,IF((U62+E62)>=0,U62+E62,F62)}/UE62~{LET
E62,IF((F62-U62)>=0,F62-U62,E62)} >>D:T.XQT
ECHO   /UF63~{LET   F63,IF((U63+E63)>=0,U63+E63,F63)}/UE63~{LET
E63,IF((F63-U63)>=0,F63-U63,E63)} >>D:T.XQT
ECHO   /UF64~{LET   F64,IF((U64+E64)>=0,U64+E64,F64)}/UE64~{LET
E64,IF((F64-U64)>=0,F64-U64,E64)} >>D:T.XQT
ECHO   /UF65~{LET   F65,IF((U65+E65)>=0,U65+E65,F65)}/UE65~{LET
E65,IF((F65-U65)>=0,F65-U65,E65)} >>D:T.XQT
ECHO /SD:RW%2,A~ >>D:T.XQT
ECHO {BEEP} >>D:T.XQT
ECHO /Q,Y~ >>D:T.XQT
CALL SC D:T
ERASE D:T.*
```

```
DIR D:/W
:END ECHO ON

rem DWOPPTNA PuT NAs in DWOPSIM product spreadsheet
rem DWOPSIM utility to put a NA in a CYPR.CAL file when PR and CN = 0
ECHO {IF D%1=0}{IF G%1=0}{LET G%1,NA}/CG%1,B%1:T%1~ >>D:T.XQT

:TIMELOOP
ECHO OFF
CLS
ECHO TIMELOOP - A batch program to loop though the TIME country list
ECHO            and call another batch program.
ECHO --------
ECHO INPUT      Batch program file name (PGNAME) called on batch
ECHO            subdirectory and any other parameters required for
ECHO            the batch program.
ECHO OUTPUT     Output of batch file called for each country/region
ECHO            in TIME.
ECHO COMMAND    TIMELOOP PGNAME (other parameters called in batch program)
ECHO --------
IF FILE%1 == FILE ECHO ERROR = You forgot ProGram NAME; Enter: TIMELOOP PGNAME
(other parameters?)
IF FILE%1 == FILE GOTO END
IF EXIST \BATCH\%1.BAT GOTO C1
ECHO \BATCH\%1.BAT does not exist
GOTO END
:C1
rem Put names of files to be run in this statement
FOR %%C IN (USBF ECBF RWBF USCN ECCN RWCN) DO CALL \BATCH\%1 TIme %%C %2 %3 %4
%5 %6
DIR D:/W
:END ECHO ON

:TIMEREPL
ECHO OFF
CLS
ECHO              TIME Replication Program for a New DWOPSIM Model
ECHO ------------------------------------------------------------------------
ECHO TIMEREPL     Program to REPLicate model specific TIME model programs and
ECHO              spreadsheet templates files for use with a new DWOPSIM
ECHO              model, NAME.  TIME is a 2 product 2 region demonstration
ECHO              model which comes with the DWOPSIM model building system.
ECHO              NAME*.BAT and NAME*.CAL files will be left on D: and must
ECHO              be edited manually to accommodate the product and country
ECHO              coverage of the new model NAME.  *.CAL and *.SUB files
ECHO              must be copied to the C:\NAME subdirectory for safekeeping.
ECHO              *.BAT files must be copied to the C:\BATCH subdirectory.
ECHO REQUIREMENTS The TIME model must exist on the C:\TIME subdirectory and
ECHO              all of the appropriate batch files must exist on the
ECHO              C:\BATCH subdirectory.
ECHO OUTPUT (D:)  The output files will reside on the D: subdirectory.
ECHO              These include DWOPNAME.BAT, NAMERW.SUB, NAMESOLV.BAT, and
ECHO              WDS.CAL (which must be customized for products in NAME).
ECHO ------------------------------------------------------------------------
ECHO COMMAND      TIMEREPL NAME
ECHO ------------------------------------------------------------------------
IF EXIST C:\BATCH\DWOPTIME.BAT GOTO C1
ECHO ERROR = The batch program DWOPTIME.BAT does not exist on C:\BATCH!
GOTO END
:C1
IF EXIST C:\BATCH\DWOPSOLV.BAT GOTO C2
ECHO ERROR = The batch program DWOPSOLV.BAT does not exist on C:\BATCH!
GOTO END
:C2
IF EXIST C:\TIME\TIMERW.SUB GOTO C3
ECHO ERROR = The TSView *.SUB file TIMERW.SUB does not exist on C:\TIME!
GOTO END
:C3
IF FILE%1 == FILE ECHO ERROR = You forgot new model NAME; Enter: TIMEREPL NAME
IF FILE%1 == FILE GOTO END
PAUSE
COPY C:\TIME\TIMERW.SUB D:%1RW.SUB
COPY C:\BATCH\DWOPTIME.BAT D:DWOP%1.BAT
COPY C:\BATCH\TIMESOLV.BAT D:%1SOLV.BAT
CALL RWORD DWOP%1.BAT TIME %1
CALL RWORD %1SOLV.BAT TIME %1
CLS
DIR D: /W
:END ECHO ON

:DWOPINIT
ECHO OFF
CLS
ECHO              DWOPSIM Program
ECHO ------------------------------------------------------------------------
ECHO DWOPINIT     Program to INITialize equations in a model spreadsheet,
ECHO              given the equation cell components.  Equations with a
ECHO              calculated constant term are inserted from the base period
ECHO              forward.  A FILE name is assumed for the model
ECHO              spreadsheets (e.g. USBF.CAL).
ECHO REQUIREMENTS A C:\NAME subdirectory must contain the model spreadsheet
ECHO              FILE and the FILEEQ equation specification spread-
ECHO              sheet.  The equation writing spreadsheet must exist on the
ECHO              C:\DWOPSIM subdirectory.
ECHO OUTPUT (D:)  The initialized spreadsheet FILE on D:.
ECHO ------------------------------------------------------------------------
ECHO COMMAND      DWOPINIT NAME FILE
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE  ECHO ERROR = You forgot model NAME; Enter: DWOPINIT NAME FILE
IF FILE%1 == FILE  GOTO END
IF FILE%2 == FILE  ECHO ERROR = You forgot spreadsheet FILE; Enter: DWOPINIT NAME
FILE
IF FILE%2 == FILE  GOTO END
```

```
IF EXIST C:\%1\%2.CAL GOTO C1
ECHO ERROR = The C:\%1\%2.CAL model spreadsheet does not exist!
GOTO END
:C1
IF EXIST C:\%1\%2EQ.CAL GOTO C2
ECHO ERROR = The C:\%1\%2EQ.CAL equation spreadsheet does not exist!
GOTO END
:C2
IF EXIST C:\DWOPSIM\EQUATION.CAL GOTO C3
ECHO ERROR = The C:\DWOPSIM\EQUATION.CAL equation writing spreadsheet does not
exist!
GOTO END
:C3
CLS
ECHO ...
ECHO Getting the files %2.CAL and %2EQ.CAL for initialization.
ECHO ...
IF EXIST D:??.XQT.CAL ERASE D:??.XQT.CAL
COPY C:\%1\%2EQ.CAL D:
COPY C:\%1\%2.CAL D:
C:\DWOPSIM\INIT
COPY C:\DWOPSIM\EQUATION.CAL D:
ECHO {MACRO} >D:T2.XQT
ECHO {STATUS "Writing out the macros to initialize the %2.CAL spreadsheet!"}
>>D:T2.XQT
ECHO {PROMPT "They will be read in and executed by the next step."} >>T2.XQT
ECHO {MESSAGE "A DWOPSIM routine by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:T2.XQT
ECHO {WINDOWSOFF} >>D:T2.XQT
ECHO {PANELOFF} >>D:T2.XQT
ECHO {ENTRYOFF} >>D:T2.XQT
ECHO /LD:EQUATION,R~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PA1,A1~ >>D:T2.XQT
ECHO /UB2~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PB2,B2~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PC1:AN4,C1~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PB6:AN43,B6~ >>D:T2.XQT
ECHO {CALC} >>D:T2.XQT
ECHO {OPEN D:V.XQT,W} >>D:T2.XQT
COPY D:T2.XQT+D:T1.XQT D:T.XQT
ECHO {MACRO} >D:R.XQT
ECHO {CALC} >>D:R.XQT
ECHO /SD:%2.CAL,OA~ >>D:R.XQT
ECHO {BEEP 2} >>D:R.XQT
ECHO /Q,Y~ >>D:R.XQT
CALL SC D:T
CALL D:T %1
ERASE D:T.*
ECHO {MACRO} >D:T.XQT
ECHO /LD:%2.CAL,R~ >>D:T.XQT
ECHO {WINDOWSOFF} >>D:T.XQT
ECHO {PANELOFF} >>D:T.XQT
ECHO {ENTRYOFF} >>D:T.XQT
ECHO {STATUS "Calculating constants and writing equations for the %2.CAL
spreadsheet!"} >>D:T.XQT
ECHO {PROMPT "The %2.CAL and linked files will be left on the D: drive."}
>>D:T.XQT
ECHO {MESSAGE "A DWOPSIM routine by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:T.XQT
ECHO /XD:V~ >>D:T.XQT
CALL SC D:T.XQT
CLS
ECHO ...
ECHO House-cleaning!  Another DWOPSIM triumph by Vernon Oley Roningen.
ECHO ...
ECHO The file %2.CAL with new equations and constants is on D:
ECHO ...
ERASE D:*.XQT
ERASE D:%2EQ.CAL
ERASE D:EQUATION.CAL
CLS
DIR D:/W
:END ECHO ON

:DWOPTITL
ECHO OFF
CLS
ECHO                DWOPSIM Program
ECHO ------------------------------------------------------------------------
ECHO DWOPTITL       Program to copy TITLes into a model spreadsheet FILE from
ECHO                the equation spreadsheet (where they are originally
ECHO                entered manually).
ECHO REQUIREMENTS   A C:\NAME subdirectory must contain the FILE spreadsheet.
ECHO                The FILE spreadsheet is also formated according to
ECHO                instructions in this program.  A DWOPTITL.CAL spreadsheet
ECHO                containing format instructions must be on the DWOPSIM
ECHO                subdirectory.
ECHO OUTPUT (D:)    The titled FILE spreadsheet on D:.
ECHO ------------------------------------------------------------------------
ECHO COMMAND        DWOPTITL NAME FILE
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE  ECHO ERROR = You forgot model NAME; Enter: DWOPTITL NAME FILE
IF FILE%1 == FILE  GOTO END
IF FILE%2 == FILE  ECHO ERROR = You forgot model FILE name; Enter: DWOPTITL NAME
FILE
IF FILE%2 == FILE  GOTO END
IF EXIST C:\%1\%2.CAL GOTO C1
ECHO ERROR = C:\%1\%2.CAL does not exist!
GOTO END
:C1
IF EXIST C:\%1\%2EQ.CAL GOTO C2
ECHO ERROR = C:\%1\%2EQ.CAL does not exist!
GOTO END
:C2
```

```
IF EXIST C:\DWOPSIM\DWOPTITL.CAL GOTO C3
ECHO ERROR = C:\DWOPSIM\DWOPTITL.CAL does not exist!
GOTO END
:C3
CLS
ECHO ...
ECHO Getting file %2.CAL and %2EQ.CAL.
ECHO ...
IF EXIST D:??.XQT.CAL ERASE D:??.XQT.CAL
COPY C:\%1\%2EQ.CAL D:
COPY C:\%1\%2.CAL D:
C:\DWOPSIM\INIT
COPY C:\DWOPSIM\EQUATION.CAL D:
ECHO {MACRO} >D:T2.XQT
ECHO {STATUS "Doing housekeeping!"} >>D:T2.XQT
ECHO {PROMPT "Be patient!"} >>T2.XQT
ECHO {MESSAGE "A DWOPSIM routine by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:T2.XQT
ECHO {WINDOWSOFF} >>D:T2.XQT
ECHO {PANELOFF} >>D:T2.XQT
ECHO {ENTRYOFF} >>D:T2.XQT
ECHO /LD:EQUATION,R~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PA1,A1~ >>D:T2.XQT
ECHO /UB2~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PB2,B2~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PC1:AN4,C1~ >>D:T2.XQT
ECHO /LD:%2EQ.CAL,PB6:AN43,B6~ >>D:T2.XQT
ECHO {CALC} >>D:T2.XQT
ECHO {OPEN D:V.XQT,W} >>D:T2.XQT
COPY D:T2.XQT+D:T1.XQT D:T.XQT
ECHO {MACRO} >D:R.XQT
ECHO {CALC} >>D:R.XQT
ECHO /Q,Y~ >>D:R.XQT
CALL SC D:T
CALL D:T %1
ERASE D:T.*
COPY C:\DWOPSIM\DWOPTITL.CAL D:
ECHO {MACRO} >D:T1.XQT
ECHO {STATUS "Writing out the column titles in the %2.CAL spreadsheet!"}
>>D:T1.XQT
ECHO {PROMPT "%2.CAl will be saved on the D: drive."} >>T1.XQT
ECHO {MESSAGE "A DWOPSIM routine by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:T1.XQT
ECHO {WINDOWSOFF} >>D:T1.XQT
ECHO {PANELOFF} >>D:T1.XQT
ECHO {ENTRYOFF} >>D:T1.XQT
ECHO /LD:%2EQ,R~ >>D:T1.XQT
ECHO /LD:DWOPTITL,A~ >>D:T1.XQT
ECHO {CALC} >>D:T1.XQT
ECHO {OPEN D:T2.XQT,W} >>D:T1.XQT
ECHO {WRITELN B53} >>D:T1.XQT
ECHO {WRITELN C53} >>D:T1.XQT
ECHO {WRITELN D53} >>D:T1.XQT
ECHO {WRITELN E53} >>D:T1.XQT
ECHO {WRITELN F53} >>D:T1.XQT
ECHO {WRITELN G53} >>D:T1.XQT
ECHO {WRITELN H53} >>D:T1.XQT
ECHO {WRITELN I53} >>D:T1.XQT
ECHO {WRITELN J53} >>D:T1.XQT
ECHO {WRITELN C54} >>D:T1.XQT
ECHO {WRITELN D54} >>D:T1.XQT
ECHO {WRITELN E54} >>D:T1.XQT
ECHO {WRITELN F54} >>D:T1.XQT
ECHO {WRITELN G54} >>D:T1.XQT
ECHO {WRITELN H54} >>D:T1.XQT
ECHO {WRITELN I54} >>D:T1.XQT
ECHO {WRITELN J54} >>D:T1.XQT
ECHO {WRITELN C55} >>D:T1.XQT
ECHO {WRITELN D55} >>D:T1.XQT
ECHO {WRITELN E55} >>D:T1.XQT
ECHO {WRITELN F55} >>D:T1.XQT
ECHO {WRITELN G55} >>D:T1.XQT
ECHO {WRITELN H55} >>D:T1.XQT
ECHO {WRITELN I55} >>D:T1.XQT
ECHO {WRITELN J55} >>D:T1.XQT
ECHO {WRITELN B55} >>D:T1.XQT
ECHO {CLOSE} >>D:T1.XQT
ECHO /LD:%2,RY~ >>D:T1.XQT
REM These lines format the columns of the country/commodity spreadsheet.
ECHO /FCB,I~ >>D:T1.XQT
ECHO /FCC,U8~ >>D:T1.XQT
ECHO /FCD:AC,I~ >>D:T1.XQT
ECHO /FEB1:AC3,TR~ >>D:T1.XQT
REM End of formatting lines.
ECHO /XD:T2~ >>D:T1.XQT
ECHO {MACRO} >D:T3.XQT
ECHO /BB1:AC3~ >>D:T3.XQT
ECHO {CALC} >>D:T3.XQT
ECHO /SD:%2,O~ >>D:T3.XQT
ECHO /Q,Y~ >>D:T3.XQT
CALL SC D:T1
ERASE D:*.XQT
ERASE D:%2EQ.CAL
ERASE D:DWOPTITL.CAL
COPY D:%2.CAL D:TT
ERASE D:??.CAL
ERASE D:%2??.CAL
COPY D:TT D:%2.CAL
ERASE D:TT
ERASE D:EQUATION.CAL
CLS
ECHO ...
ECHO The file %2.CAL is on D: with titles from %2EQ.CAL.
ECHO ...
```

```
DIR D:/W
:END ECHO ON

:DWOPBOUT
ECHO OFF
CLS
ECHO                DWOPSIM Output Program
ECHO ---------------------------------------------------------------------
ECHO DWOPBOUT       Program to print a page of a DWOPsim Base data spreadsheet
ECHO                OUTput.
ECHO REQUIREMENTS   DWOPSIM model spreadsheet with assumed FILE name (e.g.
ECHO                USBF.CAL) on NAME model subdirectory.  The template file
ECHO                DWOPBOUT.CAL must be available on the SWOPSIM subdirectory.
ECHO                The print (P) option assumes an HPLASER batch program
ECHO                calling a PORTSM97 configuration file.
ECHO OUTPUT (D:)    Print 1 page file on D: (with P option) or on D: (with F
ECHO                option) and a FILEBOUT.CAL on D:
ECHO ---------------------------------------------------------------------
ECHO COMMAND        DWOPBOUT  NAME FILE P (or F)
ECHO ---------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot model NAME; Enter: DWOPBOUT NAME FILE
P(or F)
IF FILE%1 == FILE GOTO END
IF FILE%2 == FILE ECHO ERROR = You forgot spreadsheet FILE name; Enter: DWOPBOUT
NAME FILE P(or F)
IF FILE%2 == FILE GOTO END
IF FILE%3 == FILE ECHO ERROR = You forgot Print (or File) code; Enter DWOPBOUT
NAME CY PR P(or F)
IF FILE%3 == FILE GOTO END
IF EXIST C:\DWOPSIM\DWOPBOUT.CAL GOTO C1
ECHO ERRPR = You have not created the C:\DWOPSIM\DWOPBOUT.CAL template yet!
GOTO END
:C1
IF EXIST C:\%1\%2.CAL GOTO C2
ECHO ERROR = You have not created the C:\%1\%2.CAL CountrY-PRoduct
ECHO         equation spreadsheet yet!
GOTO END
:C2
CLS
ECHO ...
ECHO The file %2.CAL for the model %1 is being called.
ECHO ...
IF EXIST D:TEST.XQT ERASE D:TEST.XQT
IF EXIST D:%2BOUT.CAL ERASE D:%2BOUT.CAL
IF EXIST D:%2BOUT.PRN ERASE D:%2BOUT.PRN
REM Call batch file to set print size with HPLASER file PORTSM97
IF F%3 == FP CALL C:HPLASER PORTSM97
COPY C:\%1\%2.CAL D:B.CAL
COPY C:\DWOPSIM\DWOPBOUT.CAL D:
REM The following statements write the SC5 macro to print out the data.
ECHO {MACRO} >D:TEST.XQT
ECHO {STATUS "Printing data from the C:\%1\%2.CAL spreadsheet."} >>D:TEST.XQT
ECHO {PROMPT "A file %2BOUT.CAL will be left on D:."} >>D:TEST.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:TEST>XQT
ECHO {PANELOFF} >>D:TEST.XQT
ECHO {WINDOWSOFF} >>D:TEST.XQT
ECHO /LD:DWOPBOUT,R~ >>D:TEST.XQT
ECHO {LET P5,%1} >>D:TEST.XQT
ECHO {CALC} >>D:TEST.XQT
ECHO /SD:%2BOUT,V~ >>D:TEST.XQT
IF F%3 == FP ECHO /OPGQ~ >>D:TEST.XQT
IF F%3 == FF ECHO /OFD:%2BOUT~GQ~ >>D:TEST.XQT
ECHO {BEEP 1} >>D:TEST.XQT
ECHO /Q,Y~ >>D:TEST.XQT
CALL C:SC D:TEST
ERASE D:TEST.XQT
ERASE D:B.CAL
ERASE D:DWOPBOUT.CAL
CLS
DIR D:*.*/W
:END ECHO ON

:DWOPEOUT
ECHO OFF
CLS
ECHO                DWOPSIM Output Program
ECHO ---------------------------------------------------------------------
ECHO DWOPEOUT       Program to print 2 pages of a DWOPsim Equation and
ECHO                parameter spreadsheet OUTput FILE as well as a
ECHO                3rd page of model equations.
ECHO REQUIREMENTS   DWOPSIM equation spreadsheet file (FILEEQ.CAL)
ECHO                on NAME model subdirectory.  The template file DWOPEOUT.CAL
ECHO                must be available on the DWOPSIM subdirectory.  The print
ECHO                (P) option assumes an HPLASER batch program calling a
ECHO                PORTSMAL configuration file for the Laser Control program.
ECHO OUTPUT (D:)    Print 2 page file on printer (with P option) or on D: (with
ECHO                F option) and a FILEEOUT.CAL on D:.  Print 1 page of model
ECHO                variable names and equations (on printer or on D:).
ECHO ---------------------------------------------------------------------
ECHO COMMAND        DWOPEOUT  NAME FILE P (or F)
ECHO ---------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot model NAME; Enter: DWOPEOUT NAME FILE
P(or F)
IF FILE%1 == FILE GOTO END
IF FILE%2 == FILE ECHO ERROR = You forgot Country code; Enter: DWOPEOUT NAME FILE
P(or F)
IF FILE%2 == FILE GOTO END
IF FILE%3 == FILE ECHO ERROR = You forgot Print (or File) code; Enter DWOPEOUT
NAME FILE P(or F)
IF FILE%3 == FILE GOTO END
IF EXIST C:\DWOPSIM\DWOPEOUT.CAL GOTO C1
ECHO You have not created the C:\DWOPSIM\DWOPEOUT.CAL template yet!
GOTO END
```

```
:C1
IF EXIST C:\%1\%2EQ.CAL GOTO C2
ECHO You have not created the C:\%1\%2EQ.CAL equation spreadsheet file yet!
GOTO END
:C2
IF EXIST C:\DWOPSIM\EQNWRITE.CAL GOTO C3
ECHO You have not created the C:\DWOPSIM\EQNWRITE.CAL equation writing
spreadsheet yet!
GOTO END
:C3
CLS
ECHO ...
ECHO The file %2EQ.CAL for the model %1 is being called.
ECHO ...
IF EXIST D:TEST.XQT ERASE D:TEST.XQT
IF EXIST D:%2EOUT.CAL ERASE D:%2EOUT.CAL
IF EXIST D:%2EOUT.PRN ERASE D:%2EOUT.PRN
REM Call batch file to set print size with Laser Control program
IF F%3 == FP CALL C:\HPLASER PORTSMAL
COPY C:\%1\%2EQ.CAL D:Q.CAL
COPY C:\DWOPSIM\DWOPEOUT.CAL D:
REM The following statements write the SC5 macro to print out the elasticities.
ECHO {MACRO} >D:TEST.XQT
ECHO {STATUS "Printing matrix of equation terms and equations for: %2"}
>>D:TEST.XQT
ECHO {PROMPT "A file %2EOUT.CAL will be left on D:."} >>D:TEST.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:TEST.XQT
ECHO {PANELOFF} >>D:TEST.XQT
ECHO {WINDOWSOFF} >>D:TEST.XQT
ECHO /LD:DWOPEOUT,R~ >>D:TEST.XQT
ECHO {LET L135,%1} >>D:TEST.XQT
ECHO {CALC} >>D:TEST.XQT
ECHO /SD:%2EOUT,V~ >>D:TEST.XQT
IF F%3 == FP ECHO /OPGQ~/Z,Y >>D:TEST.XQT
IF F%3 == FF ECHO /OFD:%2EOUT~GQ~/Z,Y >>D:TEST.XQT
ECHO /LC:\DWOPSIM\EQNWRITE,R~ >>D:TEST.XQT
ECHO /LC:\%1\%2EQ,A~{CALC} >>D:TEST.XQT
ECHO {LET K85,%1} >>D:TEST.XQT
IF F%3 == FP ECHO /OPRA85:K167~GQ~ >>D:TEST.XQT
IF F%3 == FF ECHO /OFD:%2EQN~RA85:K167~GQ~ >>D:TEST.XQT
ECHO {BEEP 1} >>D:TEST.XQT
ECHO /Q,Y~ >>D:TEST.XQT
CALL C:SC D:TEST
ERASE D:TEST.XQT
ERASE D:Q.CAL
ERASE D:DWOPEOUT.CAL
CLS
DIR D:*.*/W
:END ECHO ON
```

```
:DWOPSOLV
ECHO OFF
CLS
ECHO                DWOPSIM World Model SOLVer Assembly Program
ECHO ------------------------------------------------------------------------
ECHO DWOPSOLV       Program to assemble a DWOPSIM world model in memory and
ECHO                prepare it for SOLVing.  A macro is also created
ECHO                which can be used to save the results on D: after the
ECHO                model is solved.  Type ALT-S to invoke the 'save' macro.
ECHO                The model can be solved again with the program NAMEAGAN.
ECHO REQUIREMENTS   DWOPSIM model spreadsheets in the NAME subdirectory.  A
ECHO                1 Digit letter or number must be selected to mark the
ECHO                solution value files saved on D:  This program also
ECHO                requires that a model specific batch file has to have
ECHO                been created and saved on the batch subdirectory (under
ECHO                the name NAMESOLV.BAT).  This file tells DWOPSOLV what
ECHO                model files comprise the global model to be assembled.
ECHO                A WDS.CAL world solution mechanism file must have been
ECHO                created on the NAME subdirectory.
ECHO OUTPUT (D:)    Linked files in memory for the model NAME, ready for
ECHO                SOLVing.  If the ALT-S macro is invoked after the model is
ECHO                solved, D: will contain solution value files appended with
ECHO                the selected Digit (e.g. if D=1 for USBF, file is USBF.S1).
ECHO ------------------------------------------------------------------------
ECHO COMMAND        DWOPSOLV  NAME D
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot model NAME; Enter: DWOPSOLV NAME D
IF FILE%1 == FILE GOTO END
IF FILE%2 == FILE ECHO ERROR = You forgot 1 Digit solution marker; Enter DWOPSOLV
NAME D
IF FILE%2 == FILE GOTO END
IF EXIST C:\%1\WDS.CAL GOTO C1
ECHO You have not created the C:\%1\WDS.CAL world solution file yet!
GOTO END
:C1
IF EXIST C:\%1\WD.CAL GOTO C2
ECHO You have not created the C:\%1\WD.CAL WorlD (price) spreadsheet yet!
GOTO END
:C2
IF EXIST C:\BATCH\%1SOLV.BAT GOTO C3
ECHO You have not created the batch file %1SOLV for the model %1 on the
ECHO C:\BATCH subdirectory.  This file must be created and customized for
ECHO the model %1.
GOTO END
:C3
IF EXIST D:WD.S%2 ECHO A solution with the code %2 already exists on D:.  Try
another code!
IF EXIST D:WD.S%2 GOTO END
CLS
ECHO ...
ECHO Generating macros for assembling model %1 for solution!
ECHO ...
```

```
IF EXIST D:SS.XQT ERASE D:SS.XQT
ECHO {MACRO} >D:SS.XQT
ECHO {STATUS "Assembling the world model %1 in memory!  When done,"} >>D:SS.XQT
ECHO {PROMPT "the model can be solved by calculating the spreadsheet (F9)."}
>>D:SS.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:SS.XQT
ECHO {PANELOFF} >>D:SS.XQT
ECHO {WINDOWSOFF} >>D:SS.XQT
CALL %1SOLV %2
CALL DWOPWDS %1
ERASE D:TEST.XQT
ERASE D:SS.XQT
ERASE D:WDS.CAL
ERASE D:S.XQT
IF NOT EXIST D:WD.S%2 ERASE D:*.S%2
CLS
ECHO Model files available for another scenario are:
DIR D:*.CAL/W/P
IF EXIST D:WD.S%2 ECHO .
IF EXIST D:WD.S%2 ECHO Solution value files from this scenario are:
IF EXIST D:WD.S%2 DIR D:*.S%2/W/P
:END ECHO ON

:TIMESOLV
ECHO OFF
CLS
ECHO              DWOPSIM World Model SOLVer Model Specific Assembly Program
ECHO ------------------------------------------------------------------------
ECHO TIMESOLV     Program to assemble a DWOPSIM world model in memory and
ECHO              prepare it for SOLVing the model TIME.  This program must
ECHO              be customized for each model by repeating commands for
ECHO              each country in the model (after each REM statement).
ECHO              A macro is created to save the solution after the model
ECHO              in memory is solved.  Each file saved on D: will be the
ECHO              values from the solution file with a 1 (or 2) digit letter
ECHO              or number added.  For example, USBF.CAL will become
ECHO              USBF.S1 if 1 is chosen as the Digit to mark the solution.
ECHO REQUIREMENTS DWOPSIM Product, Country, and World spreadsheets in the
ECHO              NAME subdirectory.
ECHO OUTPUT (D:)  Linked files in memory for the model NAME, ready for
ECHO              SOLVing.  Solution value files will be saved on D: if the
ECHO              ALT S command is invoked after the model has solved.
ECHO ------------------------------------------------------------------------
ECHO COMMAND      TIMESOLV D
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot 1 Digit solution marker; Enter
TIMESOLV D
IF FILE%1 == FILE GOTO END
ECHO {MACRO} >D:T.XQT
ECHO {STATUS "Getting the equation parts of country spreadsheets and putting"}
>>D:T.XQT
ECHO {PROMPT "them on the D: drive for later insertion into memory."} >>D:T.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:T.XQT
ECHO {PANELOFF} >>D:T.XQT
ECHO {WINDOWSOFF} >>D:T.XQT
rem The following commands copy the product, country, and world files from
rem the TIME subdirectory to the D: drive.  Copy commands must exist for
rem each commodity file in the model.
rem Lines to be edited will be bracketed by REM ********** lines.
REM Copy model files to D:**************************************************
COPY C:\TIME\USBF.CAL D:
COPY C:\TIME\ECBF.CAL D:
COPY C:\TIME\RWBF.CAL D:
COPY C:\TIME\USCN.CAL D:
COPY C:\TIME\ECCN.CAL D:
COPY C:\TIME\RWCN.CAL D:
COPY C:\TIME\??.CAL D:
COPY C:\TIME\WDS.CAL D:
REM ************************************************************************
rem The following commands load country spreadsheets from D:
rem and save the equation laden parts of them on D: for use in
rem the linked world model.  A statement should be included for each country
rem in the model.  If the time structure of the model changes, the range in
rem the load statements must be changed.  Currently, assuming the base year
rem is in row 36, 9 years of lags can be handled.  All models must have an RW
rem (rest of world) component for closure, but different models can have
rem different country components.
REM Macros to load equations from spreadsheets into memory*****************
ECHO /Z,Y~/LUSBF~PA27:AC65~A27~/LUSBF~PA1:AC4~A1~/SUSBF~OA~ >>D:T.XQT
ECHO /Z,Y~/LECBF~PA27:AC65~A27~/LECBF~PA1:AC4~A1~/SECBF~OA~ >>D:T.XQT
ECHO /Z,Y~/LRWBF~PA27:AC65~A27~/LRWBF~PA1:AC4~A1~/SRWBF~OA~ >>D:T.XQT
ECHO /Z,Y~/LUSCN~PA27:AC65~A27~/LUSCN~PA1:AC4~A1~/SUSCN~OA~ >>D:T.XQT
ECHO /Z,Y~/LECCN~PA27:AC65~A27~/LECCN~PA1:AC4~A1~/SECCN~OA~ >>D:T.XQT
ECHO /Z,Y~/LRWCN~PA27:AC65~A27~/LRWCN~PA1:AC4~A1~/SRWCN~OA~ >>D:T.XQT
ECHO /Z,Y~/LUS~PA27:H65~A27~/LUS~PA1:H4~A1~/SUS~OA~ >>D:T.XQT
ECHO /Z,Y~/LEC~PA27:H65~A27~/LEC~PA1:H4~A1~/SEC~OA~ >>D:T.XQT
ECHO /Z,Y~/LRW~PA27:H65~A27~/LRW~PA1:H4~A1~/SRW~OA~ >>D:T.XQT
REM ************************************************************************
ECHO {BEEP 1}/Q,Y~ >>D:T.XQT
CALL SC D:T
ERASE D:T.XQT
CLS
ECHO ...
ECHO ...
ECHO First, backup copies of world model files are saved on D: so that
ECHO more world model scenarios can be run by the program DWOPAGAN.  When
ECHO this program is ended by saving the solution or quitting via the
ECHO //SQ command, the files save on D: with the suffix .AGN will be used
ECHO to load the model into memory for another simulation.
ECHO ...
ECHO Then, country macros are being written for the assembly of the model
ECHO TIME in memory as well as a S.XQT macro file which can be used
```

```
ECHO (press ALT S) to save the results on D: when the model is solved.
ECHO ...
ECHO ...  REMEMBER -->   ALT-S to save model results!
ECHO ...
ECHO ...
rem This section copies a set of world model files which are not kept in
rem memory as solution files.  The list copied does not include the WD file
rem but should include other files on D: which are not called into memory.
rem The files are saved on D: with the suffix *.S# where # is the 1 Digit
rem solution code.
REM DOS COPY commands to save model exogenous files as part of solution*****
COPY D:US.CAL D:US.S%1
COPY D:EC.CAL D:EC.S%1
COPY D:RW.CAL D:RW.S%1
REM ************************************************************************
rem The following commands assemble the pieces of the world model in memory.
rem Commands must exist for each country and product file to be included.
rem Files with exogenous data can be left on the D: drive.  Files with
rem variables calculated by endogenous equations must be in memory.
REM Lines to assemble pieces of world model in memory**********************
ECHO //SN~{MESSAGE "Loading USBF into memory"}/LUSBF~A~ >>D:SS.XQT
ECHO //SN~{MESSAGE "Loading ECBF into memory"}/LECBF~A~ >>D:SS.XQT
ECHO //SN~{MESSAGE "Loading RWBF into memory"}/LRWBF~A~ >>D:SS.XQT
ECHO //SN~{MESSAGE "Loading USCN into memory"}/LUSCN~A~ >>D:SS.XQT
ECHO //SN~{MESSAGE "Loading ECCN into memory"}/LECCN~A~ >>D:SS.XQT
ECHO //SN~{MESSAGE "Loading RWCN into memory"}/LRWCN~A~ >>D:SS.XQT
REM ************************************************************************
ECHO //SGN~ >>D:SS.XQT
ECHO /SWD,OA~ >>D:SS.XQT
ECHO {BEEP 2} >>D:SS.XQT
ECHO {STATUS "Now the model TIME is loaded in memory, ready to solve."}
>>D:SS.XQT
ECHO {PROMPT "Press ENTER to restore your control of the model."} >>D:SS.XQT
ECHO {MESSAGE "Press ENTER or you will be UFDAed!"} >>D:SS.XQT
ECHO {?} >>D:SS.XQT
rem Macros created in an S.XQT file to save simulation results on D:
rem The following commands create a file S.XQT on D: which can be invoked
rem by executing the ALT-S command to save the solutions values after the
rem model has been solved.
ECHO {MACRO} >D:S.XQT
ECHO {STATUS "Values of the model files are saved on D: with a 1 digit code."}
>>D:S.XQT
ECHO {PROMPT "E.g., a file USBF.CAL would have values saved as USBF.S1."}
>>D:S.XQT
ECHO {MESSAGE "A DWOPSIM SAVE by Vernon Oley Roningen, Nielsville, MN 56568."}
>>D:S.XQT
ECHO {PANELOFF} >>D:S.XQT
ECHO {WINDOWSOFF} >>D:S.XQT
ECHO /WC~ >>D:S.XQT
rem Lines of S.XQT macro for saving values of simulation results
rem These lines must exist for each spreadsheet loaded into memory above
rem plus a line for saving the world market clearing mechanism.
IF EXIST D:WD.S%1 ERASE D:*.S%1
REM Macros to save solution results*****************************************
ECHO {MESSAGE "Saving WD.S%1"}/SWD.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving USBF.S%1"}/SUSBF.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving ECBF.S%1"}/SECBF.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving RWBF.S%1"}/SRWBF.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving USCN.S%1"}/SUSCN.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving ECCN.S%1"}/SECCN.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving RWCN.S%1"}/SRWCN.S%1~V~/Q,Y~ >>D:S.XQT
REM ************************************************************************
:END ECHO ON

:DWOPWDS
ECHO OFF
CLS
ECHO              DWOPSIM WorlD price Solution mechanism addition program
ECHO ---------------------------------------------------------------------
ECHO DWOPWDS      Program to add WorlD price Solution mechanism to WD.CAL.
ECHO REQUIREMENTS DWOPSIM WD.CAL WorlD Price spreadsheet and WDS.CAL WorlD
ECHO              solution mechanism spreadsheet on C:\NAME subdirectory.
ECHO OUTPUT (D:)  WD.CAL file on D: with WorlD price Solution mechanism
ECHO              (defined in WDS.cal file) added.
ECHO ---------------------------------------------------------------------
ECHO COMMAND      DWOPWDS  NAME
ECHO ---------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot model NAME; Enter: DWOPWDS NAME
IF FILE%1 == FILE GOTO END
IF EXIST C:\%1\WDS.CAL GOTO C1
ECHO You have not created the C:\%1\WDS.CAL solution mechanism file yet!
GOTO END
:C1
IF EXIST C:\%1\WD.CAL GOTO C2
ECHO You have not created the C:\%1\WD.CAL WorlD (price) spreadsheet yet!
GOTO END
:C2
CLS
ECHO ...
ECHO Generating macros for putting solution mechanism in WD.CAL
ECHO file for model %1.
ECHO ...
IF EXIST D:TEST.XQT ERASE D:TEST.XQT
COPY C:\%1\WD.CAL D:
COPY C:\%1\WDS.CAL D:
ECHO {MACRO} >D:TEST.XQT
ECHO {STATUS "Adding world price solution mechanism to WD.CAL for model %1"}
>>D:TEST.XQT
ECHO {PROMPT "Modified file WD.CAL will be part of linked world model for %1."}
>>D:TEST.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:TEST.XQT
ECHO {PANELOFF} >>D:TEST.XQT
ECHO {WINDOWSOFF} >>D:TEST.XQT
```

```
ECHO /LWD,R~ >>D:TEST.XQT
ECHO /UA67:X102~ >>D:TEST.XQT
ECHO /LWDS,PA67:X102~A67~ >>D:TEST.XQT
ECHO {CALC} >>D:TEST.XQT
ECHO /UA103:X133~ >>D:TEST.XQT
ECHO /LWDS,PA103:X133~A103~ >>D:TEST.XQT
ECHO /UC36:X65~ >>D:TEST.XQT
ECHO /LWDS,PC36:X65~C36~ >>D:TEST.XQT
ECHO /FR103~TR~/FR69~TR~ >>D:TEST.XQT
ECHO =A1~/TC~ >>D:TEST.XQT
ECHO =A67~{DOWN 12}/WH~/WS~=A102~ >>D:TEST.XQT
ECHO /GZ~ >>D:TEST.XQT
IF EXIST D:SS.XQT ECHO /XSS~ >>D:TEST.XQT
CALL C:SC D:TEST
:END ECHO ON

:TIMEAGAN
ECHO OFF
CLS
ECHO              DWOPSIM Program to Give Another Solution for the Model TIME
ECHO ------------------------------------------------------------------------
ECHO TIMEAGAN     Program to take the world model files saved on D: by the
ECHO              program DWOPSOVE and set up the world model TIME for
ECHO              solving AGAiN. This program must be customized for each
ECHO              model by repeating commands for each country in the model
ECHO              (marked by REM lines with *****).  A solution can be saved
ECHO              and marked with a 1 digit letter or number by invoking the
ECHO              ALT S.  For example, USBF.CAL will become USBF.S1 if 1
ECHO              is chosen as the Digit to mark the solution.
ECHO REQUIREMENTS DWOPSIM Product, Country, and World spreadsheets for the
ECHO              model TIME saved on D: (e.g USBF.CAL) by the DWOPSOLV
ECHO              program.
ECHO OUTPUT (D:)  Linked files in memory for the model NAME, ready for
ECHO              SOLVing.  Solution value files will be saved on D: if the
ECHO              ALT S command is invoked after the model has solved.
ECHO ------------------------------------------------------------------------
ECHO COMMAND      TIMEAGAN D
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot 1 Digit solution marker; Enter
TIMEAGAN D
IF FILE%1 == FILE GOTO END
IF EXIST D:WD.CAL GOTO C1
ECHO There are no files for TIME on D: which are prepared by running DWOPSOLV!
GOTO END
:C1
IF EXIST D:WD.S%1 ECHO The code %1 has already been used for a solution. Try
another code!
IF EXIST D:WD.S%1 GOTO END
CLS
ECHO ...
ECHO ...
ECHO Doing the paperwork for another simulation of the model TIME
ECHO ...
ECHO ...
ECHO {MACRO} >D:T.XQT
ECHO {STATUS "Assembling the world model TIME in memory!  When done,"} >>D:T.XQT
ECHO {PROMPT "the model can be solved by calculating the spreadsheets (F9)."}
>>D:T.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:T.XQT
ECHO {PANELOFF} >>D:T.XQT
ECHO {WINDOWSOFF} >>D:T.XQT
rem The following commands assemble the world model TIME in memory.
rem Commands must exist for each country and product file to be included.
rem Files with exogenous data can be left on the D: drive.  Files with
rem variables calculated by endogenous equation must be in memory.
REM Lines to assemble pieces of world model in memory************************
ECHO /LWD,R~ >>D:T.XQT
ECHO =A1~/TC~ >>D:T.XQT
ECHO =A67~{DOWN 12}/WH~/WS~=A102~ >>D:T.XQT
ECHO /GZ~ >>D:T.XQT
ECHO //SN~{MESSAGE "Loading USBF into memory"}/LUSBF~A~ >>D:T.XQT
ECHO //SN~{MESSAGE "Loading ECBF into memory"}/LECBF~A~ >>D:T.XQT
ECHO //SN~{MESSAGE "Loading RWBF into memory"}/LRWBF~A~ >>D:T.XQT
ECHO //SN~{MESSAGE "Loading USCN into memory"}/LUSCN~A~ >>D:T.XQT
ECHO //SN~{MESSAGE "Loading ECCN into memory"}/LECCN~A~ >>D:T.XQT
ECHO //SN~{MESSAGE "Loading RWCN into memory"}/LRWCN~A~ >>D:T.XQT
REM ************************************************************************
ECHO //SGN~ >>D:T.XQT
ECHO {STATUS "Now the model TIME is loaded in memory, ready to solve."} >>D:T.XQT
ECHO {PROMPT "F9 will start the solution after exogenous shocks have been set."}
>>D:T.XQT
ECHO {MESSAGE "Press ENTER to gain control of TIME!"} >>D:T.XQT
ECHO {PANELOFF} >>D:T.XQT
ECHO {WINDOWSOFF} >>D:T.XQT
ECHO {BEEP 1} >>D:T.XQT
ECHO {?} >>D:T.XQT
rem This section copies a set of world model files which are not kept in
rem memory as solution files.  The list copied does not include the WD file
rem but should include other files on D: which are not called into memory.
rem The files are saved on D: with the suffix *.S# where # is the 1 Digit
rem solution code.
REM DOS COPY commands to save model exogenous files as part of solution*****
COPY D:US.CAL D:US.S%1
COPY D:EC.CAL D:EC.S%1
COPY D:RW.CAL D:RW.S%1
REM ************************************************************************
rem Macros created in an S.XQT file to save simulation results on D:
rem The following commands create a file S.XQT on D: which can be invoked
rem by executing the ALT-S command to save the solutions values after the
rem model has been solved.
ECHO {MACRO} >D:S.XQT
ECHO {STATUS "Values of the model files are saved on D: with a 1 digit code."}
>>D:S.XQT
```

```
ECHO {PROMPT "E.g., a file USBF.CAL would have values saved under USBF.S1."}
>>D:S.XQT
ECHO {MESSAGE "A DWOPSIM SAVE by Vernon Oley Roningen, Nielsville, MN 56568."}
>>D:S.XQT
ECHO {PANELOFF} >>D:S.XQT
ECHO {WINDOWSOFF} >>D:S.XQT
ECHO /WC~ >>D:S.XQT
rem Lines of S.XQT macro for saving values of simulation results
rem These lines must exist for each spreadsheet loaded into memory above
rem plus a line for saving the world market clearing mechanism.
IF EXIST D:WD.S%1 ERASE D:*.S%1
REM Macros to save solution results*****************************************
ECHO {MESSAGE "Saving WD.S%1"}/SWD.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving USBF.S%1"}/SUSBF.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving ECBF.S%1"}/SECBF.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving RWBF.S%1"}/SRWBF.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving USCN.S%1"}/SUSCN.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving ECCN.S%1"}/SECCN.S%1~V~/Q,Y~ >>D:S.XQT
ECHO {MESSAGE "Saving RWCN.S%1"}/SRWCN.S%1~V~/Q,Y~ >>D:S.XQT
REM *************************************************************************
CALL SC D:T.XQT
ERASE D:T.XQT
ERASE D:S.XQT
IF NOT EXIST D:WD.S%1 ERASE D:*.S%1
CLS
ECHO Model files available for another scenario are:
DIR D:*.CAL/W/P
IF EXIST D:WD.S%1 ECHO .
IF EXIST D:WD.S%1 ECHO Solution value files from this scenario are:
IF EXIST D:WD.S%1 DIR D:*.S%1/W/P
:END ECHO ON

:DWOPSOUT
ECHO OFF
CLS
ECHO                DWOPSIM Output Program
ECHO ------------------------------------------------------------------------
ECHO DWOPSOUT       Program to print a page of a DWOPsim Solution from a model
ECHO                spreadsheet OUTput FILE.
ECHO REQUIREMENTS   DWOPSIM model spreadsheet solution FILE on model
ECHO                subdirectory (NAME).  The solution spreadsheet will have
ECHO                the suffix *.SD where D is 1 Digit solution marker.
ECHO                A spread model name (FILE) is assumed.  The template file
ECHO                DWOPSOUT.CAL must be available on the SWOPSIM subdirectory.
ECHO                The print (P) option assumes an HPLASER batch program
ECHO                calling a PORTSM97 configuration file for the Laser Control
ECHO                program.
ECHO OUTPUT (D:)    Print 1 page file on D: (with P option) or on D: (with F
ECHO                option) and a FILESD.CAL on D:
ECHO ------------------------------------------------------------------------
ECHO COMMAND        DWOPSOUT  NAME FILE D P (or F)
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot model NAME; Enter: DWOPSOUT NAME FILE
D P(or F)
IF FILE%1 == FILE GOTO END
IF FILE%2 == FILE ECHO ERROR = You forgot spreadsheet FILE name; Enter: DWOPSOUT
NAME FILE D P(or F)
IF FILE%2 == FILE GOTO END
IF FILE%3 == FILE ECHO ERROR = You forgot 1 Digit solution code; Enter: DWOPSOUT
NAME FILE D P(or F)
IF FILE%3 == FILE GOTO END
IF FILE%4 == FILE ECHO ERROR = You forgot Print (or File) code; Enter DWOPSOUT
NAME FILE D P(or F)
IF FILE%4 == FILE GOTO END
IF EXIST C:\DWOPSIM\DWOPSOUT.CAL GOTO C1
ECHO ERROR = You have not created the C:\DWOPSIM\DWOPSOUT.CAL template yet!
GOTO END
:C1
IF EXIST C:\%1\%2.S%3 GOTO C2
ECHO ERROR = You have not created the C:\%1\%2.S%3 CountrY-PRoduct
ECHO         Solution spreadsheet yet!  Run DWOPSOLV!
GOTO END
:C2
CLS
ECHO ...
ECHO The solution file %2.S%3 for the model %1 is being called.
ECHO ...
IF EXIST D:TEST.XQT ERASE D:TEST.XQT
IF EXIST D:%2S%3.CAL ERASE D:%2S%3.CAL
IF EXIST D:%2S%3.PRN ERASE D:%2S%3.PRN
REM Call batch file to set print size with HPLASER file PORTSM97
IF F%4 == FP CALL C:HPLASER PORTSM97
COPY C:\%1\%2.S%3 D:S.CAL
COPY C:\%1\%2.CAL D:B.CAL
COPY C:\DWOPSIM\DWOPSOUT.CAL D:
REM The following statements write the SC5 macro to print out the data.
ECHO {MACRO} >D:TEST.XQT
ECHO {STATUS "Printing data from the C:\%1\%2.S%3 solution spreadsheet."}
>>D:TEST.XQT
ECHO {PROMPT "A file %2S%3.CAL will be left on D:."} >>D:TEST.XQT
ECHO {PANELOFF} >>D:TEST.XQT
ECHO {WINDOWSOFF} >>D:TEST.XQT
ECHO /LD:DWOPSOUT,R~ >>D:TEST.XQT
ECHO {LET A52,%3} >>D:TEST.XQT
ECHO {LET O52,%1} >>D:TEST.XQT
ECHO {CALC} >>D:TEST.XQT
ECHO /SD:%2S%3,V~ >>D:TEST.XQT
IF F%4 == FP ECHO /OPGQ~ >>D:TEST.XQT
IF F%4 == FF ECHO /OFD:%2S%3~GQ~ >>D:TEST.XQT
ECHO {BEEP 1} >>D:TEST.XQT
ECHO /Q,Y~ >>D:TEST.XQT
CALL C:SC D:TEST
ERASE D:TEST.XQT
ERASE D:B.CAL
```

```
ERASE D:S.CAL
ERASE D:DWOPSOUT.CAL
CLS
DIR D:*.*/W
:END ECHO ON

:DWOPCVAR
ECHO OFF
CLS
ECHO                DWOPSIM Output Program
ECHO ------------------------------------------------------------------------
ECHO DWOPCVAR       Program to Compare up to 4 solution VALues with baseline
ECHO                and historical values for a selected spreadsheet variable
ECHO                (Column) in a model spreadsheet FILE (e.g. USBF).
ECHO REQUIREMENTS   DWOPSIM spreadsheet and solution spreadsheet(s) on NAME
ECHO                model subdirectory.  The template file DWOPCVAR.CAL
ECHO                must be available on the DWOPSIM subdirectory.
ECHO                If the printout is invoked from the output spreadsheet,
ECHO                an HPLASER batch program calling a PORTSMAL configuration
ECHO                file for the Laser Control program is assumed available.
ECHO OUTPUT (D:)    Comparisons on screen (spreadsheet with graph in it)
ECHO ------------------------------------------------------------------------
ECHO COMMAND        DWOPCVAR  NAME FILE C S1 S2 S3 S4
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot model NAME; Enter: DWOPCVAR NAME FILE
C S1 S2 S3 S4
IF FILE%1 == FILE GOTO END
IF FILE%2 == FILE ECHO ERROR = You forgot FILE name; Enter: DWOPCVAR NAME FILE
C S1 S2 S3 S4
IF FILE%2 == FILE GOTO END
IF FILE%3 == FILE ECHO ERROR = You forgot Column letter for variable; Enter:
DWOPCVAR NAME FILE C S1 S2 S3 S4
IF FILE%3 == FILE GOTO END
IF FILE%4 == FILE ECHO ERROR = You forgot Solution digit code; Enter DWOPCVAR
NAME FILE C S1 S2 S3 S4
IF FILE%4 == FILE GOTO END
IF EXIST C:\DWOPSIM\DWOPCVAR.CAL GOTO C1
ECHO ERROR = You have not created the C:\DWOPSIM\DWOPCVAR.CAL template yet!
GOTO END
:C1
IF EXIST C:\%1\%2.CAL GOTO C2
ECHO ERROR = You have not created the C:\%1\%2.CAL model spreadsheet!
GOTO END
:C2
IF EXIST C:\%1\%2.S%4 GOTO C3
ECHO ERROR = You have not saved the C:\%1\%2.S%4 solution spreadsheet.
GOTO END
:C3
CLS
ECHO ...
ECHO The files %2.CAL and %2.S%4, etc., for the model %1 are being called.
ECHO ...
IF EXIST D:TEST.XQT ERASE D:TEST.XQT
IF EXIST D:%2V%3.CAL ERASE D:%2V%3.CAL
COPY C:\%1\%2.CAL D:A.CAL
COPY C:\%1\%2.S%4 D:B.CAL
IF NOT FILE%5 == FILE COPY C:\%1\%2.S%5 D:C.CAL
IF NOT FILE%6 == FILE COPY C:\%1\%2.S%6 D:D.CAL
IF NOT FILE%7 == FILE COPY C:\%1\%2.S%7 D:E.CAL
CALL HPLASER PORTSMAL
COPY C:\DWOPSIM\DWOPCVAR.CAL D:
REM The following statements write the SC5 macro to make the comparisons.
ECHO {MACRO} >D:TEST.XQT
ECHO {STATUS "Comparing data on the variable in column %3 in sheet %2"}
>>D:TEST.XQT
ECHO {PROMPT "from model %1 for solutions %4, %5, %6, %7."} >>D:TEST.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:TEST.XQT
ECHO {PANELOFF} >>D:TEST.XQT
ECHO {WINDOWSOFF} >>D:TEST.XQT
ECHO /LD:DWOPCVAR,R~ >>D:TEST.XQT
ECHO {STATUS "Comparing data on the variable in column %3 in sheet %2"}
>>D:TEST.XQT
ECHO {PROMPT "from model %1 for solutions %4, %5, %6, %7."} >>D:TEST.XQT
ECHO {MESSAGE "'Alt F5, Enter' will print the data and graph!"} >>D:TEST.XQT
ECHO {LETC C3,%4} >>D:TEST.XQT
ECHO {LET K1,%1} >>D:TEST.XQT
IF NOT FILE%5 == FILE ECHO {LETC D1,%5}/CC38:C67,D38~ >>D:TEST.XQT
IF NOT FILE%6 == FILE ECHO {LETC E1,%6}/CC38:C67,E38~ >>D:TEST.XQT
IF NOT FILE%7 == FILE ECHO {LETC F1,%7}/CC38:C67,F38~ >>D:TEST.XQT
ECHO {LETC G69,%3} >>D:TEST.XQT
ECHO {CALC} >>D:TEST.XQT
ECHO {BEEP 1} >>D:TEST.XQT
CALL C:SC D:TEST
ERASE D:TEST.XQT
ERASE D:A.CAL
ERASE D:B.CAL
ERASE D:DWOPCVAR.CAL
CLS
DIR D:*.*/W
:END ECHO ON

:DWOPCSOL
ECHO OFF
CLS
ECHO                DWOPSIM Output Program
ECHO ------------------------------------------------------------------------
ECHO DWOPCSOL       Program to Compare up to 4 variables in a SOLution with
ECHO                historical/baseline values.  Variables (Columns) are in
ECHO                a model FILE (e.g. USBF) and solution values are in a
ECHO                solution file (e.g. USBF.SD) where D marks the solution.
ECHO REQUIREMENTS   DWOPSIM model spreadsheet and solution spreadsheet(s) on
ECHO                model subdirectory.  The template file DWOPCSOL.CAL
ECHO                must be available on the DWOPSIM subdirectory.
```

```
ECHO              Printing (an option from the comparison spreadsheet)
ECHO              assumes an HPLASER batch program calling a PORTSMAL
ECHO              configuration file for the Laser Control program.

ECHO OUTPUT (D:)  Comparisons on screen (spreadsheet with graph in it)
ECHO ------------------------------------------------------------------
ECHO COMMAND      DWOPCSOL  NAME FILE D C1 C2 C3 C4
ECHO ------------------------------------------------------------------
IF FILE%1 == FILE ECHO ERROR = You forgot model NAME; Enter: DWOPCSOL NAME FILE
D C1 C2 C3 C4
IF FILE%1 == FILE GOTO END
IF FILE%2 == FILE ECHO ERROR = You forgot FILE name; Enter: DWOPCSOL NAME FILE
D C1 C2 C3 C4
IF FILE%2 == FILE GOTO END
IF FILE%3 == FILE ECHO ERROR = You forgot Solution code; Enter: DWOPCSOL NAME
FILE D C1 C2 C3 C4
IF FILE%3 == FILE GOTO END
IF FILE%4 == FILE ECHO ERROR = You forgot Column letter for variable; Enter
DWOPCSOL NAME FILE D C1 C2 C3 C4
IF FILE%4 == FILE GOTO END
IF EXIST C:\DWOPSIM\DWOPCSOL.CAL GOTO C1
ECHO ERROR = You have not created the C:\DWOPSIM\DWOPCSOL.CAL template yet!
GOTO END
:C1
IF EXIST C:\%1\%2.CAL GOTO C2
ECHO ERROR = You have not created the C:\%1\%2.CAL baseline data spreadsheet.
GOTO END
:C2
IF EXIST C:\%1\%2.S%3 GOTO C3
ECHO ERROR = You have not saved the C:\%1\%2.S%3 solution spreadsheet.
GOTO END
:C3
CLS
ECHO ...
ECHO The files %2.CAL and %2.S%3, etc., for the model %1 are being called.
ECHO ...
IF EXIST D:TEST.XQT ERASE D:TEST.XQT
IF EXIST D:%2S%3C%4.CAL ERASE D:%2S%3C%4.CAL
COPY C:\%1\%2.CAL D:A.CAL
COPY C:\%1\%2.S%3 D:B.CAL
CALL HPLASER PORTSMAL
COPY C:\DWOPSIM\DWOPCSOL.CAL D:
REM The following statements write the SC5 macro to make the comparisons.
ECHO {MACRO} >D:TEST.XQT
ECHO {STATUS "Comparing data for variables in solution %3 in sheet %2"}
>>D:TEST.XQT
ECHO {PROMPT "from model %1 for variables %4, %5, %6, %7."} >>D:TEST.XQT
ECHO {MESSAGE "A DWOPSIM program by Vernon Oley Roningen, Nielsville, MN 56568"}
>>D:TEST.XQT
ECHO {PANELOFF} >>D:TEST.XQT
ECHO {WINDOWSOFF} >>D:TEST.XQT
ECHO /LD:DWOPCSOL,R~ >>D:TEST.XQT
ECHO {STATUS "Comparing data for variables in solution %3 in sheet %2"}
>>D:TEST.XQT
ECHO {PROMPT "from model %1 for variables %4, %5, %6, %7."} >>D:TEST.XQT
ECHO {MESSAGE "'Alt-F5 Enter' will print the data and graph!"} >>D:TEST.XQT
ECHO {LETC A1,%1} >>D:TEST.XQT
ECHO {LETC A2,%3} >>D:TEST.XQT
ECHO {LETC A3,%2} >>D:TEST.XQT
ECHO {LETC B73,%4} >>D:TEST.XQT
IF NOT FILE%5 == FILE ECHO {LETC D73,%5} >>D:TEST.XQT
IF NOT FILE%6 == FILE ECHO {LETC F73,%6} >>D:TEST.XQT
IF NOT FILE%7 == FILE ECHO {LETC H73,%7} >>D:TEST.XQT
IF NOT FILE%5 == FILE ECHO {LETC D1,%5}/CC38:C67,D38~ >>D:TEST.XQT
IF NOT FILE%6 == FILE ECHO {LETC E1,%6}/CC38:C67,E38~ >>D:TEST.XQT
IF NOT FILE%7 == FILE ECHO {LETC F1,%7}/CC38:C67,F38~ >>D:TEST.XQT
ECHO {LETC G69,%3} >>D:TEST.XQT
ECHO {CALC} >>D:TEST.XQT
ECHO /SD:%2S%3C%4~V~ >>D:TEST.XQT
ECHO {BEEP 1} >>D:TEST.XQT
CALL C:SC D:TEST
ERASE D:TEST.XQT
ERASE D:?.CAL
ERASE D:DWOPCSOL.CAL
CLS
DIR D:*.*/W
:END ECHO ON

:DWOPTRND
ECHO OFF
CLS
ECHO              DWOPSIM Program
ECHO ------------------------------------------------------------------------
ECHO DWOPTRND     Program to put a variable (from COLumn of a spreadsheet
ECHO              FILE from the model subdirctory NAME) into a TReND
ECHO              regression spreadsheet.
ECHO REQUIREMENTS A C:\NAME subdirectory must contain the spreadsheet FILE.
ECHO OUTPUT (D:)  On screen (or printed/saved by macro in sheet on screen).
ECHO ------------------------------------------------------------------------
ECHO COMMAND      DWOPTRND NAME FILE COL
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE  ECHO ERROR = You forgot subdirectory NAME; Enter: DWOPTRND
NAME FILE COL
IF FILE%1 == FILE  GOTO END
IF FILE%2 == FILE  ECHO ERROR = You forgot spreadsheet FILE name: Enter: DWOPTRND
NAME FILE COL
IF FILE%2 == FILE  GOTO END
IF FILE%3 == FILE  ECHO ERROR = You forgot COLumn code; Enter: DWOPTRND NAME FILE
COL
IF FILE%3 == FILE  GOTO END
IF EXIST C:\%1\%2.CAL GOTO C1
ECHO ERROR = The spreadsheet C:\%1\%2.CAL does not exist!
GOTO END
:C1
```

```
IF EXIST C:\DWOPSIM\TREND.CAL GOTO C2
ECHO ERROR = C:\DWOPSIM\TREND.CAL does not exist!
GOTO END
:C2
IF EXIST D:T.XQT ERASE D:T.XQT
ECHO {MACRO}>D:T.XQT
ECHO /LC:\DWOPSIM\TREND,R~ >>D:T.XQT
ECHO /BI6:J16~ >>D:T.XQT
ECHO {LETCONTENTS J2,%1} >>D:T.XQT
REM  Change 35 to another number if you have more years of data!
ECHO /LC:\%1\%2,P%35:%335,B5,V~ >>D:T.XQT
ECHO /LC:\%1\%2,PA1,A1~ >>D:T.XQT
ECHO /LC:\%1\%2,P%33,B3~ >>D:T.XQT
ECHO /LC:\%1\%2,P%32,H3~ >>D:T.XQT
ECHO {CALC} >>D:T.XQT
CALL HPLASER TREND
CALL SC D:T
ERASE D:T.XQT
DIR D:/W
:END ECHO ON

:DWOPOLSR
ECHO OFF
CLS
ECHO                DWOPSIM Program
ECHO ------------------------------------------------------------------------
ECHO DWOPOLSR       Program to put a dependent variable (from a COLumn of a
ECHO                SpreadSheet from the Dwopsim model subdirectory NAME) along
ECHO                with up to 3 Independent variables (from Columns of
ECHO                Spreadsheets from NAME) into an Ordinary Least Squares
ECHO                Regression.
ECHO REQUIREMENTS   A C:\NAME subdirectory must contain the spreadsheet SS and
ECHO                the Spreadsheets with the independent variables.
ECHO                Specification using the 3 variables, or combinations
ECHO                including a time trend and lags, can be chosen on the
echo                screen.  Regression is done manually in SuperCalc 5.
ECHO OUTPUT (D:)    On screen (or printed/saved by macro on screen).
ECHO ------------------------------------------------------------------------
ECHO COMMAND        DWOPOLSR NAME SS COL S1 C1 S2 C2 S3 C3 S4 C4
ECHO ------------------------------------------------------------------------
IF FILE%1 == FILE   ECHO ERROR = You forgot model subdirectory NAME; Enter:
DWOPOLSR NAME SS COL S1 C2 S2 C2 S3 C3 S4 C4
IF FILE%1 == FILE   GOTO END
IF FILE%2 == FILE   ECHO ERROR = You forgot SpreadSheet name: Enter: DWOPOLSR
NAME SS COL S1 C1 S2 C2 S3 C3 S4 C4
IF FILE%2 == FILE   GOTO END
IF FILE%3 == FILE   ECHO ERROR = You forgot COLumn code for dependent variable;
Enter: DWOPOLSR NAME SS COL S1 C1 S1 C2 S3 C3 S4 C4
IF FILE%3 == FILE   GOTO END
IF FILE%4 == FILE   ECHO ERROR = You forgot Spreadsheet of independent variable;
Enter: DWOPOLSR NAME SS COL S1 C1 S1 C2 S3 C3 S4 C4
IF FILE%4 == FILE   GOTO END
IF EXIST C:\%1\%2.CAL GOTO C1
ECHO ERROR = The spreadsheet C:\%1\%2.CAL does not exist!
GOTO END
:C1
IF EXIST C:\DWOPSIM\REGRESS.CAL GOTO C2
ECHO ERROR = C:\DWOPSIM\REGRESS.CAL does not exist!
GOTO END
:C2
IF EXIST D:T.XQT ERASE D:T.XQT
ECHO {MACRO}>D:T.XQT
ECHO /LC:\DWOPSIM\REGRESS,R~ >>D:T.XQT
ECHO /BI6:J16~ >>D:T.XQT
ECHO {LETCONTENTS J2,%1} >>D:T.XQT
REM  Change 35 to another number if you have more years of data!
ECHO /LC:\%1\%2,P%35:%335,B5,V~ >>D:T.XQT
ECHO /LC:\%1\%2,PA1,A1~ >>D:T.XQT
ECHO /LC:\%1\%2,P%33,B3~ >>D:T.XQT
ECHO /LC:\%1\%2,P%32,H3~ >>D:T.XQT
ECHO /LC:\%1\%4,P%55:%565,AA5,V~ >>D:T.XQT
ECHO /LC:\%1\%4,P%53,AA3~ >>D:T.XQT
ECHO /LC:\%1\%2EQ,PB2,H17~ >>D:T.XQT
IF FILE%7 == FILE GOTO C3
ECHO /LC:\%1\%6,P%75:%765,AB5,V~ >>D:T.XQT
ECHO /LC:\%1\%6,P%73,AB3~ >>D:T.XQT
:C3
IF FILE%8 == FILE GOTO C4
ECHO /LC:\%1\%8,P%95:%965,AC5,V~ >>D:T.XQT
ECHO /LC:\%1\%8,P%93,AC3~ >>D:T.XQT
:C4
ECHO {CALC} >>D:T.XQT
CALL HPLASER TREND
CALL SC D:T
ERASE D:T.XQT
DIR D:/W
:END ECHO ON
```

The regression programs DWOPTRND and DWOPOLSR are batch programs which call template spreadsheets and insert variables from a DWOPSIM model. The user must invoke the regression commands in the SuperCalc spreadsheet to do the calculations. The user also has to check to see that the ranges set for the regression variables are consistent with the number of observations of data as well as the number of explanatory variables used. In DWOPOLSR, the equation specification can be log or linear (controlled by a transformation code in DWOPOLSR).

Reports you can use . . . from ERS

U.S. Agricultural Trade Update

gives you up-to-the-minute information.

Each month the *U.S. Agricultural Trade Update* brings you ERS' most up-to-the-minute data on the farm trade sector. This useful 6-page update brings you the most current figures, *delivered by first-class mail to ensure timely delivery.*

The *U.S. Agricultural Trade Update* covers the monthly farm trade balance, U.S. farm imports and exports by quantity and value, and leading exports and exporters.

A 1-year subscription to the *U.S. Agricultural Trade Update* costs just $15. Or save by ordering a 2-year subscription (that's 24 issues) for $29, or a 3-year subscription for $42. Please add 10% for shipment to domestic addresses; 35% for shipment to foreign addresses (includes Canada).

Situation and Outlook Agricultural Trade Reports

give you the facts . . . and the forecasts!

These reports provide both current intelligence and historical data on international food and agricultural developments. They also forecast how changes in conditions and policies around the world will affect both U.S. and international agriculture.

Outlook for U.S. Agricultural Exports offers the latest value and volume of U.S. farm exports, by commodity and region, as well as the agricultural trade balance, import commodities, and export outlook. Subscription includes 4 issues. ***Agriculture and Trade*** reports summarize the year's developments affecting U.S. agriculture and trade in five key regions, and look to the future with articles on market trends, trade, and policy (subscription includes 5 regional reports: USSR, China, Western Europe, Pacific Rim, and Asia.

The cost is just $12 for a 1-year subscription per title. Or save by ordering a 2-year subscription for $23, or a 3-year subscription for $33. Please add 10% for shipment to domestic addresses; 35% for postage to foreign addresses (includes Canada).

Foreign Agricultural Trade of the United States

gives you the latest information on U.S. markets.

This periodical updates the quantity and value of U.S. farm exports and imports, plus price trends. Its coverage of over 200 countries and more than 500 commodities will keep you informed of how U.S. trade stacks up in a global market. Subscription includes 6 issues plus calendar and fiscal year supplements.

The cost is just $25 for a 1-year subscription. Or save by ordering a 2-year subscription for $49, or a 3-year subscription for $72. Please add 10% for shipment to domestic addresses; 35% for shipment to foreign addresses (includes Canada).

Order now! Rates good through 12/92.

Call our order desk toll free, 1-800-999-6779
in the U.S. and Canada (other areas, please call 703-834-0125).
Or write: ERS-NASS, 341 Victory Drive, Herndon, VA 22070.

Back issues of ERS periodicals in prepackaged sets

Fill in your collection of ERS periodicals with packaged sets of our valuable data and analysis, now sale priced. While supplies last, ERS is offering sets of back issues of some of its major periodicals. Round out your personal collection of *Agricultural Outlook*, for example, with the complete set of issues from 1990 or 1991. Or find the historical data you need on farm income and expenses, production and efficiency statistics, and more with past years' issues of *Economic Indicators of the Farm Sector*.

It's easy! Just write ERS-NASS at 341 Victory Drive, Herndon, VA 22070, **or call 1-800-999-6779** (in the U.S. and Canada; other areas call 703-834-0125) today.

And attention libraries: We offer an additional **$5 discount per periodical package** to your institution. Just submit your order on letterhead, or on a purchase order with the pre-printed name and address of your library. **Please** note the discounted price in your order.

Agricultural Outlook '90. 11 issues; regular price if sold separately, $88.
Order # PKG-18. $25.

Agricultural Outlook '91.
11 issues; regular price if sold separately, $88.
Order # PKG-19. $25.
(Sorry, 1989 Agricultural Outlook not available.)

Foreign Agricultural Trade of the United States, '89.
6 issues; regular price if sold separately, $48.
Order # PKG-20. $20.

Foreign Agricultural Trade of the United States, '90.
6 issues; regular price if sold separately, $48.
Order # PKG-21. $20.

Foreign Agricultural Trade of the United States, '91.
6 issues; regular price if sold separately, $48.
Order # PKG-22. $20.

Farmline '89.
11 issues; regular price if sold separately, $88.
Order # PKG-23. $15.

Farmline '90.
11 issues; regular price if sold separately, $88.
Order # PKG-24. $15.

Farmline '91.
11 issues; regular price if sold separately, $88.
Order # PKG-25. $15.

Economic Indicators of the Farm Sector, '88.
5 issues covering 1988 statistics; regular price if sold separately, $40.
Order # PKG-26. $18.

Economic Indicators of the Farm Sector, '89.
5 issues covering 1989 statistics; regular price if sold separately, $40.
Order # PKG-27. $18.

Food Review '89.
4 issues; regular price if sold separately, $32.
Order # PKG-28. $16.

Food Review '90.
4 issues; regular price if sold separately, $32.
Order # PKG-29. $16.

Food Review '91.
4 issues; regular price if sold separately, $32.
Order # PKG-30. $16.

Journal of Agricultural Economics Research, '89.
4 issues; regular price if sold separately, $32.
Order # PKG-31. $15.

Journal of Agricultural Economics Research, '90.
4 issues; regular price if sold separately, $32.
Order # PKG-32. $15.

Journal of Agricultural Economics Research, '91.
4 issues; regular price if sold separately, $32.
Order # PKG-33. $15.

Order now! Prices good through December 1992.
Add 10% for shipping to domestic addresses; 35% to foreign addresses.

U.S. Department of Agriculture
Economic Research Service
1301 New York Avenue, NW.
Washington, DC 20005-4788.